Kingston University

Thesis

Scene Analysis and Risk Estimation for Domestic Robots, Security and Smart Homes

Author:
Rob Dupré

Supervisor:
Dr. Vasilis Argyriou
Dr. Darrel Greenhill

in the

Digital Information Research Centre
Faculty of Science, Engineering and Computing

February 2017

This Thesis is being submitted in partial fulfilment of the requirements of Kingston University for the Degree of Doctor of Philosophy (Ph.D.)

Copyright Statement

Declaration of Authorship

- This report is submitted as requirement for a Ph.D. Degree in the School of Computing and Information Systems (Faculty of Science, Engineering and Computing) at Kingston University. It is substantially the result of my own work except where explicitly indicated in the text.
- No portion of the work referred to in this report has been submitted in support of an application for another degree or qualification of this or any other UK or foreign examination board, university or other institute of learning.
- The thesis work was conducted from October 2013 to September 2016 under the supervision of Dr Vasileios Argyriou and Dr Darrel Greenhill in the Digital Information Research Centre (DIRC) of Kingston University, London.

KINGSTON UNIVERSITY

Abstract

Faculty of Science, Engineering and Computing

Doctor of Philosophy

Scene Analysis and Risk Estimation for Domestic Robots, Security and Smart Homes

by Rob Dupré

The evaluation of risk within a scene is a new and emerging area of research. With the advent of smart enabled homes and the continued development and implementation of domestic robotics, the platform for automated risk assessment within the home is now a possibility. The aim of this thesis is to explore a subsection of the problems facing the detection and quantification of risk in a domestic setting.

A Risk Estimation framework is introduced which provides a flexible and context aware platform from which measurable elements of risk can be combined to create a final risk score for a scene. To populate this framework, three elements of measurable risk are proposed and evaluated: Firstly, scene stability, assessing the location and stability of objects within an environment through the use of physics simulation techniques. Secondly, hazard feature analysis using two specifically designed novel feature descriptors (3D Voxel HOG and the Physics Behaviour Feature) which determine if the objects within a scene have dangerous or risky properties such as blades or points. Finally, environment interaction, which uses human behaviour simulation to predict human reactions to detected risks and highlight areas of a scene most likely to be visited.

Additionally methodologies are introduced to support these concepts including: a simulation prediction framework which reduces the computational cost of physics simulation, a Robust Filter and Complex Adaboost which aim to improve the robustness and training times required for hazard feature classification models. The Human and Group Behaviour Evaluation framework is introduced to provide a platform from which simulation algorithms can be evaluated without the need for extensive ground truth data. Finally the 3D Risk Scenes (3DRS) dataset is introduced, creating a risk specific dataset for the evaluation of future domestic risk analysis methodologies.

Acknowledgements

The work within the thesis would not have been possible without the support of my supervisor; Dr Vasileios Argyriou and Dr Darrel Greenhill. I offer my sincerest gratitude for their patience, advice and guidance over the last three years. Their input on the work for this thesis and my own personal development has been invaluable.

I wish to express my thanks to the members (past and present) of the Digital Information Research Centre. Their support when I first undertook this journey was so important to my research as was the discussion of new and existing ideas in a open and inviting environment. The welcoming nature of the group is a testament to the members and the supervisory team behind it. I specifically wish to mention Dr Dimitrios Makris, Dr Jean-Christophe Nebel, Dr Francisco Florez Revuelta, Dr Andreas Hoppe, Hope Caton and Dr Gordon Hunter for their guidance and critical insight. Also; Dr Spyros Bakos, Dr Victoria Bloom and Dr Pau Climent Pérez for their friendship and guidance throughout my PhD.

I would like to extend my thanks to Kingston University for funding my research, as well as their partners (specifically NVIDIA) for their provision of vital hardware and software enabling us to continue the work we do. I also wish to extend my thanks to the Kingston University post graduate support team who's provisioning of training and administrative support throughout my time was a huge help.

Finally to my family; my mother and father, who quite literally made me the man I am today and my sisters who take great delight in keeping my feet firmly on the ground.

Contents

List of Figures

List of Tables

Abbreviations

2D	**2 D**imensions
2.5D	**2.5 D**imensions
3D	**3 D**imensions
3D VHOG	**3D V**oxel **HOG**
3DRS	**3D R**isk **S**cenes
BRDF	**B**idirectional **R**eflectance **D**istribution **F**unction
BTF	**B**idirectional **T**exture **F**unctions
BU-3DFE	**B**inghamton **U**niversity - **3D F**acial **E**xpression
CCTV	**C**losed-**C**ircuit **T**ele**V**ision
cm	**c**enti**m**eter
CNN	**C**onvolutional **N**eural **N**etwork
CPMC	**C**onstrained **P**arametric **M**id-**C**ut
CPU	**C**entral **P**rocessing **U**nit
EM	**E**xpectation **M**aximization
EMD	**E**arth **M**over's **D**istance
FAST	**F**eatures from **A**ccelerated **S**egment **T**est
FPFH	**F**ast **P**oint **F**eature **H**istograms
fps	**f**rames **p**er **s**econd
HC	**H**yper **C**omplex
HOG	**H**istogram of **O**rientated **G**radients
HOOF	**H**istogram of **O**rientated **O**ptical **F**low
HVS	**H**uman **V**isual **S**ystem
ICA	**I**ndependent **C**omponent **A**nalysis
IFR	**I**nternational **F**ederation of **R**obotics
IPE	**I**ntuitive **P**hysics **E**ngine

LiDAR	**L**ight **D**etection **A**nd **R**adar
MCOV	**M**ulti-Scale **COV**ariance
MEU	**M**aximum **E**xpected **U**tility
MLE	**M**aximum **L**ikelihood **E**stimation
MOS	**M**ean **O**pinion **S**core
MRI	**M**agnetic **R**esonance **I**maging
ms	**m**illi**s**econds
NYU	**N**ew **Y**ork **U**niversity
PAL	**P**hase **A**lternating **L**ine
PBF	**P**hysics **B**ehaviour **F**eature
PCR	**P**rincipal **C**omponent **R**egression
PCA	**P**rincipal **C**omponent **A**nalysis
PETS	**P**erformance **E**valuation of **T**racking **S**urveillance
PFH	**P**oint **F**eature **H**istograms
RBK	**R**oyal **B**orough **K**ingston
RGB	**R**ed **G**reen **B**lue
RGB-D	**R**ed **G**reen **B**lue - **D**epth
SA	**S**ituational **A**wareness
SFM	**S**ocial **F**orce **M**odel
SHOT	**S**ignature of **H**istogram **O**rien**T**ations
SIFT	**S**cale **I**nvariant **F**eature **T**ransform
SLAM	**S**imultaneous **L**ocalisation **A**nd **M**apping
STD	**ST**andard **D**eviation
SURF	**S**peeded **U**p **R**obust **F**eatures
SV-DHDP	**S**tochastic **V**ariational **D**ual **H**ierarchical **D**irichlet **P**rocess
SVM	**S**upport **V**ector **M**achine
SVR	**S**upport **V**ector Machines for **R**egression

Symbols

R	Risk score
e	Element which measures risk
S	Stability measurement (overall)
w_S	Weighting for a instability measurement
s	Instability for a locally applied force
F	Force
K	Kinetic Energy
$\mathbf{P}$	Probability
δx	Object displacement
M	Mass of an object
V	Velocity
T	Time
x, y, z	Components of position in 3D cartesian space
$\mathbf{Y}$	Dependent variables within a PCR dataset
$\mathbf{X}$	Independent variables within a PCR dataset
$\mathbf{B}$	Model, PCR or Adaboost
ϵ	error variable
$\mathbf{V}$	Matrix of eigenvalues
Λ	Matric of eigenvectors
W	Low dimensional representation of $\mathbf{X}$
γ	Least squares regression
H	Hazard features
w_H	Weighting for a hazard features measurement
ω	Angular velocity
f	Feature block

c	Cell contained within a feature block
v	voxel
$\vec{g}$	Gradient vector
$\|\vec{g}\|$	Magnitude of the gradient vector
w	Weighting
h	Histogram
θ	Angle around 360^o
ϕ	Angle around 180^o
ε	Smallest number > 0
C	Adaboost or Complex Adaboost classification
d	Robust filter representation of given value
α	Robust filter variable for controlling frequency of the cosine
β	Weak classifier for an Adaboost model
D	Importance weight for complex Adaboost weak classifier
Z	Normalisation factor based on ϵ
J	Training iterations
Q	Cumulative integral image
Q_{Dim}	Multidimensional integral image
f	Column index of integral image
c	Row index of integral image
X^p	Multidimensional rectangle sample
E	Environmental risk map risk element
w_E	Weighting for Interaction map risk element
P	Positions in terms of x and y, and later ν
$\mathbf{H}$	Summed histogram for environmental risk maps
$\mathbf{F}$	Field of view
a	Subject agent
ν	Defined visibility value based on $\mathbf{F}$
q	Viewable radius from an agent
g	Destination of an agent
p	Position of an agent in 2D space
b	Detected agents in the proximity of the subject agent a
o	Detected obstacles in the proximity of the subject agent a

$\mathbf{S}$	State in decision network
A	Action that can be performed in a given state
U	Utility, measure of benefit from a given action A
EU	Expected utility
$\mathbf{E}$	Evidence, knowledge that a given agent bases their decisions on
B	Background
I	Image from video sequence, pixel intensities as a specific time
n	number of frames in a video sequence
$\mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{l}$	User defined points for perspective plane
$\mathbf{u}_1, \mathbf{u}_2$	User defined point of measure for perspective plane
$\mathbf{T}$	Automatically defined points as a result of plane extraction
$\mathbf{T}_{vanish}$	Automatically defined vanishing point used in plane extraction
G	Scaled points for definition of the perspective plane grid
m	Defined unit of measure from which perspective scale is based
$\mathbf{R}$	Random automatically defined points in plane extraction
dm	Differential change in motion
dV	Differential change in velocity
L	Control variable for use in HVS features with Weber's Law
$\mathbf{M}$	Motion
$\vec{u}$	Motion vector at a time t
$M_R,\ M_S$	Motion vectors from real and simulated sequences
$T_R,\ T_S$	Tracklets from real and simulated sequences
$f_R^{HOOF},\ f_S^{HOOF}$	Histogram or Orientated Optical Flow, real and simulated frames
$f_R^{H2D},\ f_S^{H2D}$	2D HOOF, real and simulated sequences
$f_R^{T},\ f_S^{T}$	Compounded Tracklet image for real and simulated sequences
Φ	Human visual system features

Publications

Some ideas and figures have appeared previously in the following publications:

Journals:

- **R. Dupre**, V. Argyriou, G. Tzimiropoulos and D. Greenhill, 'Risk analysis for smart homes and domestic robots using robust shape and physics descriptors and complex boosting techniques,' *Information Sciences*, vol. 372, pp. 359-379, 2016.
- **R. Dupre**, V. Argyriou and D. Greenhill, 'Prediction of physics simulation using dimensionality reduction and regression,' *Journal of Graphics Tools*, vol. 17, no. 3, pp. 99-110, 2013.
- E. Konstantinidis, A. Billis, **R. Dupre**, J. Fernández Montenegro, G. Conti, V. Argyriou and P. Bamidis, 'IoT of active and healthy ageing: cases from indoor location analytics in the wild,' *Journal of Health and Technology*, pp. 1-9, 2016

Conference Papers:

- **R. Dupre** and V. Argyriou, 'Risk assessment for RGBD scans in real time,' *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2084-2088, 2016.
- **R. Dupre**, V. Argyriou, D. Greenhill, and G. Tzimiropoulos, 'A 3D scene analysis framework and descriptors for risk evaluation,' *2015 International Conference on 3D Vision (3DV)*, pp. 100-108, 2015.
- **R. Dupre** and V. Argyriou, '3D Voxel HOG and risk estimation,' *2015 IEEE International Conference on Digital Signal Processing (DSP)*, pp. 482-486, 2015.

- **R. Dupre** and V. Argyriou, 'Prediction of physics simulations for graphics and animation,' *Proceedings of SIGRAD 2014 Visual Computing*, vol. 106, pp. 83-86, 2014.
- **R. Dupre**, R. Acuna, V. Argyriou and S. Velastin, '3D Interaction environment for free view point TV and games using multiple tablet computers,' *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 682-687, 2013.

Under Review:

- **R. Dupre**, V. Argyriou and D. Greenhill, 'Automated environmental risk analysis through scene evaluation and human behaviour simulation,' *IEEE Transactions on Robotics.*
- **R. Dupre**, V. Argyriou and D. Greenhill, 'Pedestrian and crowd simulation analysis using compositing and video comparison,' *Computer Graphics Forum.*

Chapter 1

Introduction

Scene analysis is the problem of describing the contents of a picture of a three dimensional scene [18] and was pioneered by the work of Roberts in 1965 [19]. Although the techniques have evolved, the goal has largely remained the same over the last 50 years. To broaden the definition slightly: scene analysis is the definition of context or extraction of knowledge from a given environment via computer vision techniques. This provides the ability to define systems that can provide us with the information to make decisions on a given situation. As an example, the identification of humans within a scene or the recognition of objects. The number of applications is growing and with the introduction of ever more accurate and commercially viable hardware [20] this technology is becoming far more accessible.

One such application for scene analysis is the assessment of risk. A risk can be considered as a situation that could harm someone or something (Figure 1.1). Within our day to day lives we, as humans, are exposed to risks almost constantly. As these risks are encountered, they are identified and appropriate action is taken to keep ourselves safe. However for certain members of society these abilities are diminished and low level

FIGURE 1.1: Risky situations

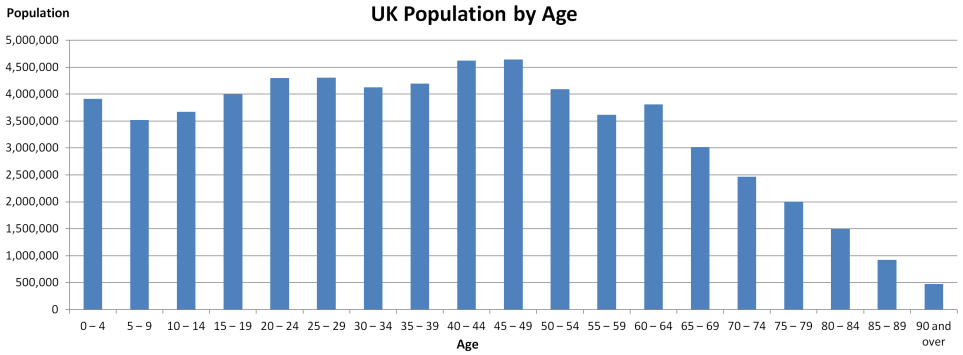


FIGURE 1.2: Graph of the distribution of the UK population by age, based on the 2011 Census [1].

hazards can become far more of a problem or even endanger lives. For example, most people will make use of a kitchen in their daily routine and the act of leaving a knife near the edge of the sideboard would not pose a significant risk (Figure 1.3). Consider now the same scenario but with an elderly adult suffering from Parkinson's disease or early onset dementia being the user of the kitchen, or with a child running around the room. The possibility of knocking that knife on the floor for a healthy adult is low, but for a child not actively aware of their surroundings or those suffering from a disability this could become more of a likelihood.

Invariably there is a percentage of society that could be classed as more *at risk* than the rest (Figure 1.4). There are an estimated 10 *million* disabled people living in the UK [21] and the elderly (65+ years) and young children (< 10 years) account for 28% of the UK population [1] (Figure 1.2). People within this age range are statistically more likely to have an accident in the home [22]. Additionally the number of people that fall into the elderly category is increasing. Europe has one of largest aging populations with 24% already aged 60 years or over, this is set to increase sharply such that by 2050 that proportion is projected to reach 34% [23]. As such the infrastructure which provides continued support services to this population will come under increasing pressure. With this increase in the amount of elderly adults continuing to live self sufficiently, the need to alleviate the stresses on the support services and the need to provide a high level of care, emerging technologies can be used to ensure they remain safe and if something were to happen, appropriate services can be notified or actions taken.

As such the following work proposes the notion that risk as a concept can be measured providing a numerical scaling that would allow thresholds of risk to be defined and appropriate actions taken. This work would facilitate the ability for domestic robots or

FIGURE 1.3: Example situation with a knife left precariously at the edge of a table

smart enabled homes to detect potential risks, for example ensuring that a room is safe for a child.

The advent of the smart home is once such example of applicable technology that can be used to aid this problem. Indeed it is already in use to help the elderly and disabled live safe and independent lives [24]. Smart homes consist of distributed sensor networks throughout a residence, examples include Closed Circuit Television(CCTV), depth cameras, heat sensors or thermal imaging cameras as well as the advent of smart assistants and other automated decision making support devices. By combining the outputs of these sensor networks an analysis of the environment is possible.

Not all homes are smart homes and although the retro fitting of many of these products is possible, it is not always viable. Domestic robotics provides an alternative to this through the use of service robots. A service robot has the purpose of either aiding or performing actions that contribute toward the improvement of the quality of life of an individual [25]. Indeed the International Federation of Robotics (IFR) have noted a steady increase in the sales of professional and personal robotics since 2012, with predictions that during the period of 2014-2107 the estimated number of service robots sold for domestic/personal use could be as high as 27 million [26]. As such the research and development in the field of domestic robotics is gaining momentum, looking into the functional hardware and software required to create these devices. Research is also being conducted into the expectations and attitudes that we as humans would have to such devices [27], as well as the operational safety requirements for use around humans [25, 28]. Utilising these to the issue of risk evaluation is an obvious one, as it provides

FIGURE 1.4: Potential users and environments in which risk detection would be beneficial

the option to not only identify and draw attention to the risks, but also to potentially take action to prevent them.

Risk can take many forms and as such the work proposed here looks to create methods that identify areas of risk and provide quantifiable risk scores from which actions can be taken. These areas of risk include: the stability of objects in a scene, the hazardousness of the objects themselves and impact human interaction has to the objects. To this end a risk framework is introduced that provides an open ended solution for combining these different elements of risk and producing a final risk score for an environment. To identify risk a number of different disciplines are utilised: these range from physics simulation and scene analysis to human behaviour simulation. Each of these methods have been utilised to define a specific area of risk: object stability, detection of hazardous features such as knife blades or sharp corners and finally the impact human behaviour has on risk in an environment and inversely the affect risk has on human behaviour.

Risk itself is a contextual problem. A risk may effect one group of people more so than others. As well as the type of risk, the environment in which the risk is found is also relevant to the hazard that risk might pose. For example a container of liquid at the edge of a table might well pose a risk, however that risk is dramatically increased if the environment in which that container sits is a lab and the liquid was a highly corrosive acid. This idea of contextual awareness requires the proposed solution to be adaptable to individual circumstances, requiring the ability to scale the various risk measurements so as to take into account both those that use the environment and the environment itself.

1.1 Aims and Objectives

Given the issues already outlined, a number of aims can be defined from which this work seeks to address. The first of which being the development of a Risk Estimation Framework to produce a quantifiable risk score for any given environment. To this end the definition of the framework is required as well as the ability to measure various elements of risk. In this case these are the measurement of object stability, hazardous properties of an object and the impact that human behaviour has on risk.

As a result of these aims more specific objectives can be derived. These objectives are given below in a their developmental order:

- Define a extendable framework capable of utilising any element of measurable risk and which takes into consideration the context in which the risk is analysed.
- Utilise simulation techniques to measure the stability of an object in its environment from captured scenes using depth sensors.
- Address the issue of the high computational cost involved with existing simulation techniques.
- Develop methods that can evaluate the hazardous properties ('hazard features') of an object from captured scenes using depth sensors.
- Evaluate ways in which these methods can be improved and made robust.
- Provide the facility to simulate human behaviour towards risk in an environment and provide robust methods by which they can be evaluated.
- Create a risk specific dataset in which scenes and objects can be evaluated in the context of risk assessment.

1.2 Contributions to Knowledge

The first contribution is the definition of the extendable Risk Estimation framework [15], in which the facility is provided to utilise measurable forms of risk and apply a context based weighting to tailor the results to the given situation. Importantly the framework

must be extendable, providing the basis by which any future risk measurements can be included. A risk specific dataset is also presented; 3D Risk Scenes(3DRS)[15] dataset containing individual objects and scenes with both hazardous and safe household objects, which provides a challenging dataset from which risk evaluation can be performed. Data is provided in both synthetic models (computer aided design based), as well as reconstructed scenes using commercially available depth sensing hardware. Additionally frame by frame sequences of depth scans are also available allowing methodologies to be evaluated in real time data samples.

Secondly the contribution of object and scene stability [29] derived using a novel combination of machine learning and physics simulation techniques. Using the resultant energy outputs due to an applied force in a simulation environment and, taking into account the subsequent effects on other objects within a scene, a picture of object stability can be formed. This allows assertions to be made about the total stability of that scene. As an example consider the difference between a glass bottle placed at the corner of a table, against it being placed at the centre. In addition a novel dynamic approach to physics simulation using machine learning is presented [30, 31]. Through the use of machine learning and dimensionality reduction techniques, the computational requirements needed for such processes are dramatically reduced. Using the proposed methodology complex physics scenarios can be learned and in future predicted to reduce the overall complexity and computational workload with only a marginal reduction in accuracy. This contribution also stands as the basis for work into the concept of prediction as a tool within scene analysis and computer vision.

Thirdly feature descriptor methodologies are presented using novel 3D shape descriptors and Newtonian physics based on supervised learning. The 3D Voxel HOG (3D VHOG) descriptor [15, 32] tries to identify dangerous elements or characteristics of an object (e.g. a knife's blade being sharp, pointed). Here object recognition is not the goal allowing the approach to be more general and operate at a lower level. The Physics Behaviour Feature (PBF) descriptor [29] is outlined utilising the physical properties of an object to identify if it is hazardous. The descriptor makes use of data produced using simulation techniques based on Newtonian Physics and the estimation of an object's angular velocity after the application of a force. With both these methods a boosting technique (Adaboost [33]), results in a model which aims to specify whether an object affects the potential risk in a given 3D scene.

Fourthly the introduction of the novel Robust Filter [34] for 3D descriptors which looks to reduce the effect of outliers in the machine learning process to ultimately produce more accurate and robust models from training data. As a result of this an extension to the original Adaboost [33] algorithm is presented in the form of Complex and Hyper-Complex variants [34]. Leveraging the properties of complex and hyper complex numbers to provide an increase in computational efficiency of model generation.

Finally methods to compute the effect that human behaviour has on risk are presented. A novel simulation algorithm that emulates a humans response to a risk is introduced and used to build up interaction maps that are used as another element of risk in the Risk Estimation framework. In evaluating the effectiveness of this methodology a framework has been suggested that aims to measure the similarity of pedestrian and crowd simulation algorithms, by comparing source video data and a new video sequence created using simulation data and compositing techniques. Characteristics of the Human Visual System (HVS) are utilised to create novel evaluation techniques. This provides tangible and relevant metrics which can be used for a quantitative comparison between simulation algorithms as well as simulation tuning.

1.3 Structure of the Thesis

In Chapter 2, an overview of the literature and background information that is important in the context of this thesis is presented. Initially scene analysis research with regard to hazard and risk is reviewed along with key concepts with regard to data processing and data description. Relevant machine learning and physics simulation processes are analysed. The area of human behaviour simulation and simulation evaluation techniques are scrutinized and finally a overview of the datasets utilised within this thesis is given.

The risk evaluation framework and the first proposed element of risk is outlined in Chapter 3. Preprocessing techniques used in the preparation of the scene analysis data is explained and the methodology for stability estimation of objects within a scene is introduced. Finally the prediction mechanism, by which the computational requirements of stability estimation is reduced, is presented. Evaluation results for each of the methodologies is given and reviewed using the 3D Risk Scenes dataset.

Within Chapter 4 the second risk element pertaining to the object hazard detection is presented. Two hazard feature descriptors (3D Voxel HOG and the Physics Behaviour Feature) are introduced which detect potentially dangerous properties of objects such as the presence of sharp corners or blades. With the use of machine learning and filtering techniques, robust models are created by which classification can be performed. Evaluation of the methodologies is performed using the 3D Risk Scenes dataset. Additionally the 3D Voxel HOG feature is evaluated on the popular computer vision problem of facial expression recognition using the BU-3DFE dataset.

Chapter 5 introduces the last proposed element of risk in the form of environmental risk maps. Using a novel human behaviour simulation model, human interaction with an environment is estimated and used to build an interaction assessment map. By analysing and applying the likelihood of human interaction with detected risks in an environment a more complete picture is created of how hazardous that risk is. Additionally a evaluation framework is presented which aims to reduce the complexity of analysing simulation algorithms. Through the use of compositing and visualisation tools, simulation evaluation can be performed using only source video with no need for comprehensive ground truth data. An extensive evaluation is given using a range of crowd simulation datasets.

Finally in Chapter 6 conclusions and intended future work is outlined.

Chapter 2

Literature Review

The following chapter will concern itself with similar work within the areas of scene analysis, hazard and risk related research, human behaviour simulation and other relevant concepts appropriate to the work in the following chapters. In each case any pertinent and, where possible, state of the art methods are discussed and analysed. Additionally any techniques or methodologies utilised within this thesis are also outlined.

2.1 Scene Analysis

To enable the estimation of risk in an environment, that environment must first be analysed. Scene analysis is a broad area of computer vision which aims to provide the functionality to achieve these goals. These techniques enable the determination of context or extraction of information from a given scene. Within scene analysis there are two notable areas of research: The first, 2D, where information is extracted from an image or video sequence. The second is 3D, where the use of RGB-D cameras or other depth capturing hardware obtains a three dimensional representation of an environment that is then analysed for a given purpose.

Principally the work discussed going forward will be focused on the 3D side of scene analysis, however, as much of the existing work is based on techniques defined for the 2D environment, certain seminal works will be briefly discussed. As such a comprehensive review of the key areas will be given.

2.1.1 Data Acquisition and Preprocessing

For risks in a scene to be analysed, a digital representation of it must first be obtained. In the realms of two dimensions this tends to be through the use of an image sensor that measures light intensity. This will either be in a single image or as an image sequence or video.

As the direction of this work is intended to be in 3D, the data acquisition methods looked at will focus on those techniques that better represent this domain. The goal of these types of data acquisition techniques is to produce a data structure that overcomes the limitations of 2D, such as occlusions, segmentation, background extraction. This can be achieved in a number of ways, the most common being RGB-D data. This concept was first used with the introduction of Microsoft's Kinect sensor [20]. Kinect augments the standard RGB data from an image sensor with an additional data layer representing the distance a pixel is from the sensor plane. This is not a true representation of 3D and as such is often referred to as 2.5D. Figure 2.1 demonstrates the two inputs (RGB and depth) and an example point cloud which can be produced as a result.

With the advent of commercially available depth acquisition hardware [20] and laser scanning systems such as LiDAR, scene analysis research in the 3D domain has grown considerably [35–37]. Using these types of hardware, techniques such as Simultaneous Localisation and Mapping (SLAM) [38] provide the ability to construct a mesh model of a scene by moving the capture hardware through the environment. SLAM combines the sensor input information with the pose information of the capture device to map each frame of data into a fully 3D environment. This data is stitched together so as to produce a working 3D model of an environment (Figure 2.2), Kinect Fusion [4] being a well known example of this technique. Other SLAM techniques make use of multiple static cameras as opposed to a single camera moved through an environment. After effective calibration, 3D scenes can be constructed without the need to compensate for an ever changing camera pose [39].

The outputs of these devices usually manifest in a number of ways; the first is separate data layers such as RGB images and depth images. Another often used data type constructed using these two data sources is the point cloud. Point clouds are a volumetric data representation of an environment where each point has a three dimensional location and potentially some other property such as its colour. Extending the point clouds

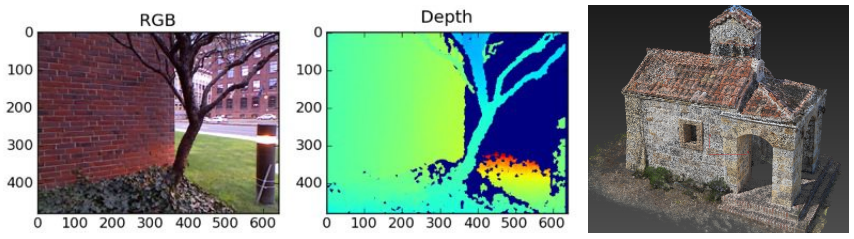


FIGURE 2.1: Examples of the RDB-D data capture process and an example point cloud with coloured points [2, 3]

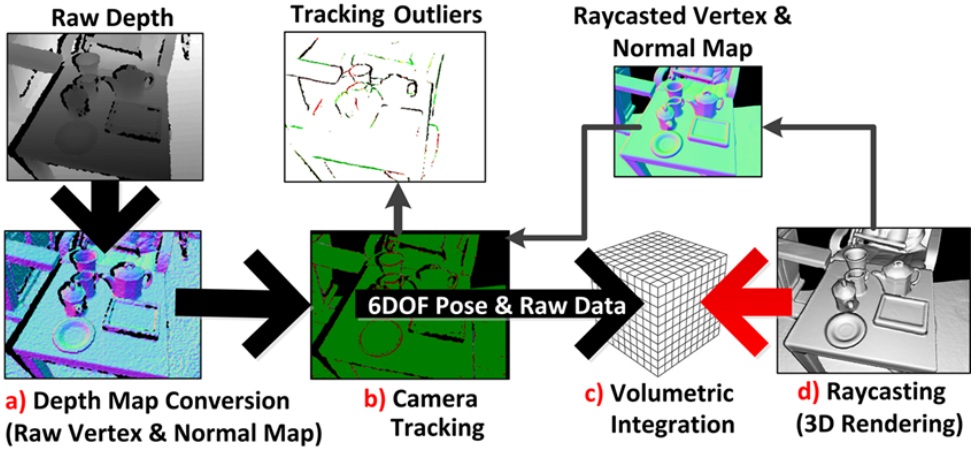


FIGURE 2.2: Kinect Fusion pipeline [4]

volumetric property, Voxelization [40] allows 3D scene data to be represented by an evenly spaced grid based volume in which each grid reference in 3D space (a voxel) is represented by a value.

Once this 3D volume data has been defined, further preprocessing techniques can be used to remove unwanted data or add additional context. For example Trevor et al [41] removes surfaces, such as table tops, from within point clouds using connected components and a 'planar refinement step'. This allows for better clustering of the objects of interest within the scene. Defining bounding information for objects in a scene has also become a valid area of research, providing the functionality to quantify object properties such as size and position [42]. To establish the physical parameters of a scene object, such as mass or friction coefficients, identification of its material is preferable but not always practical and instead can be estimated. Material recognition has had a number of approaches suggested; [43] makes use of a number of different features combined using a Bayesian generative framework which allows the method to learn what the optimum set of features are. Material recognition has the major challenge of differentiating texture from material, i.e there could be three objects that have a similar pattern that are made from entirely different materials. Another suggested approach

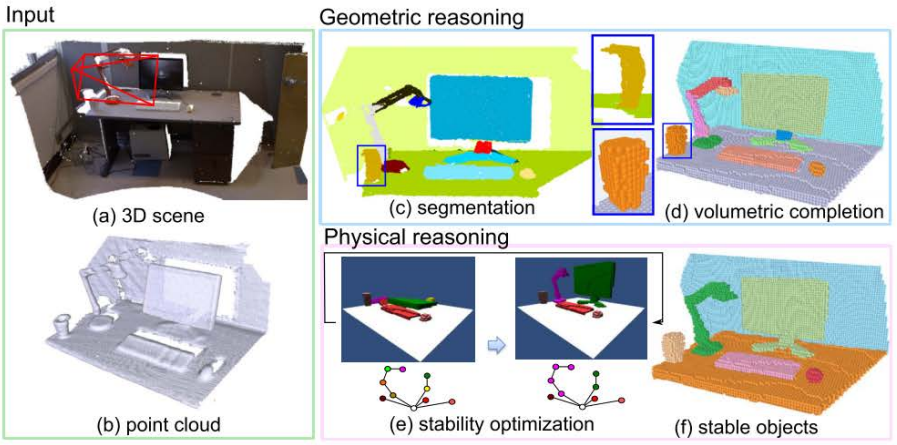


FIGURE 2.3: Pipeline as presented within [5], from data acquisition to volumetric reasoning

by Liu et al [44] uses features from Bidirectional Texture Functions (BTF) and takes a learning approach to define illumination patterns and filters for use in classification problems. Through the use of material estimation a better understanding of the objets within a scene can be garnered. For example the weight of an object can make up a potential element of risk. Additionally the extra data can make physics simulation more accurate and therefore more relevant.

2.1.2 Object Segmentation and Clustering

As the scene that is analysed needs to be separated into its individual components to better facilitate risk analysis, segmentation is an essential task. The field of segmentation has had much research in the 2D domain and more recently in the 3D space.

Image segmentation in 2D can be described as the partitioning of an image into objects of interest [45]. Although much work has been done into this specific problem, a number of recent innovations are given as examples. The work in [46] uses a grid system and constrained parametric mid-cut problems (CPMC) to segment an image. This is coupled with a ranking system to determine how plausible the object segmentation is likely to be. Another 2D technique [45] uses a combination of background subtraction with edge detection filters to define masks for possible objects.

Similarly to 2D, 3D segmentation aims to partition the data. This is especially important with 3D data as being able to eliminate areas of data or focus the analysis on small subsets of the data improves processing time. In the realms of risk evaluation this is important to the creation of a system that is functional in the domestic environment.

The area of 3D segmentation can be approached using a number of methodologies. One such example is feature based classification, where high level context is applied to define objects or areas, such as the work in [47] on the NYU Depth data set. Here rich features are extracted at a low level and a linear Support Vector Machine (SVM) is used to define a classifier. This combined with kernel descriptors provides segmentation of the scene.

Alternatively clustering is used, in which parts of the scene data (patterns, points or objects) are naturally grouped. An overview of the clustering topic is discussed in [48] in which a comprehensive look at the last 50 years of research is analysed. Clustering has a huge number of different categories, defined not only by how they function but also what parameters are defined from the start as well as the cluster definition output. Liu et al. [49], suggested an effective use of a fuzzy C Means algorithm to segregate a 3D planar object map, while remaining unsupervised. Within the work of Do et al [50], K- Means clustering is used as part of their method to reduce degradation of 3D reconstructions of occluded objects, combined with independent component analysis (ICA).

Another emerging area of research within scene analysis relates to 3D volumetric reasoning. This area concerns itself with the concept of identifying object volumes in an environment, their stability, and if that volume is supported by others within the scene. This draws heavily from a human's ability to analyse a scene and make fast judgements about the environment. Battaglia [51] explores this concept and introduces the idea of a 'Intuitive Physics Engine (IPE)' which tries to mimic a human's cognitive simulation process when analysing a scene. Wu et al [52] extends this principle by incorporating a physics engine with respect to learning. Their work further supports the idea that a human's ability to analyse a scene is based upon a realistic physics engine as part of a generative model to interpret real-world physical scenes. Additionally the system is also capable of outputting physical properties of objects from video observations such as mass and friction coefficients.

Zheng et al [5] utilises the notion that clusters in a scene should be at a state of rest when simulation techniques are applied. For example a cluster containing the voxels of a computer screen should also have those that make up its stand included as part of the defined object (Figure 2.3). Using an iterative process, clusters are grouped until the scene is at equilibrium. To achieve this a physics engine is utilised to model the application of physical principles, such as gravity, to the object clusters.

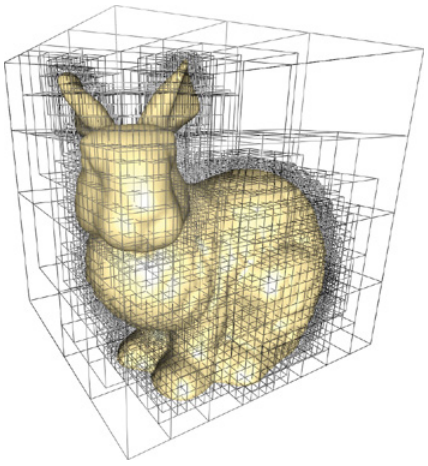


FIGURE 2.4: Example 3D model with a surrounding octTree [6]

Jia et al [42] proposes a similar method that better fits bounding shapes to RGB-D point cloud clusters. This is based on the premise that a good 3D representation of a scene is stable, fits the data well and is self-supporting. Using supporting relations and the stability of the scene given the bounding shapes, segmentation can be improved using a learning process to better fit the boxes to the point cloud.

These 3D blocks can be better defined through the use of bounding box refinement techniques such as octrees [53]. These allow for more accurate bounding shapes to be assigned to objects within a scene by constructing a shape comprised of various sized cubes (Figure 2.4). This provides a bounding shape that ensures all the parts of the object are inside the shape whilst ensuring as little empty space as possible is also included.

2.2 Risk Evaluation in Static Scenes

The concept of measuring risk in a scene is a relatively new area of research. Existing techniques for risk in financial markets exist [54], however the problem of evaluating physical risk in an environment remains largely unanswered.

Zheng et al [11, 16], evaluate risk in a scene through the analysis of the probability that an object could be dislodged. Through the use of disturbance fields, human interaction as well as natural disturbance, such as wind or the effect of earthquakes, is modeled to create risk scores for objects within the scene. Using this data the probability of objects falling can be calculated. This yields a risk score based on a specific type of input, which

requires modeling per event. However the approach does not take into account the risk associated with the objects themselves, the effects the objects will have on the rest of the environment or the possibility that objects may collide with each other when disturbed. Nor does it take into account the cognitive abilities of the humans that interact with the environment.

Other work on risk assessment exists in similar areas such as patient monitoring [55, 56], where the focus is on indoor fall assessment for elderly adults. Here monitoring techniques are used to try and determine if the subject has had an accident or whether the properties associated with how they walk through an environment might indicate something is wrong. Though conceptually similar, these papers focus on analysing the risk associated with the person and not their environment. Work on robotics for medical applications by Dannenmann et al [57] defines safety zones around anatomical areas, such as major nerval and vascular structures. This prevents the robotic system entering these zones, providing an efficient way of preventing injury and localising potential risks. However, the system does not apply reasoning to the environment. Additionally although the system tracks patients movement, it requires pre-programming for each change in situation.

With advances in the industrial robotic sector and robotic hardware, new areas of risk in various workplaces have been identified. In [58, 59], a review is provided into these hazards and the principles of guarding to ensure human safety. Hazard analysis, safety precautions, programming procedures and maintenance of the robots are also discussed.

Finally, with advances in robotics and unmanned drones, the functionality to fully automate these devices using vision based techniques is emerging [60, 61]. Though these proposed systems do not emphatically determine risk, they do analyse the environment to identify a suitable landing zone based on a set of parameters.

2.2.1 Feature Descriptors

To facilitate the analysis of risk in a static scene, methods are required which firstly allow the localisation of interest points; secondly, the description of specific scene properties and lastly, allow the recognition of these properties in other scenes. Detection,

description and recognition are all research subjects that have received a huge amount of work in recent years both in the 2D and 3D domains.

The feature detector is concerned with finding locations presenting rich visual information and whose spatial location is well defined [62], which will be used as the focus or subject of the final feature vector. Some common example properties in the image domain might be corners, ridges, edges or blobs / regions of interest. As this often forms the basis for the subsequent feature descriptor, it is important that whatever aspect of the data being detected, it should be repeatable in other examples of similar data. It should be noted that a detector is not always required, some feature descriptor methods simply iterate over all the data in one form or another. Feature detection is used in the preliminary steps of many computer vision problems such as tracking, simulations localisation and mapping (SLAM), image matching and recognition.

Feature description is the characterizing of a local property within the data, through the creation of a vector (numerical representation) [62]. What can be represented can be almost anything and will depend largely on how the data was captured. Recognition is concerned with the identification of these features in other scenes.

For complex scene analysis tasks more specific feature descriptors are required. For example, defining the presence of a bicycle in an image. To achieve this a descriptor is required which converts the image data of a bicycle into a feature vector. Using the same conversion process, new images can have the same process applied creating new undefined feature vectors. With the use of a matching process, the new unknown feature vector can be compared to the original and its similarity measured.

How to define this feature is an ongoing area of research in computer vision. Two primary questions exist in this area; firstly, what part of the data should be used to construct the feature? The edges of the bike, the whole bike image, just the wheels etc. Secondly; what is the best way of representing this data as a feature? These same issues exist when defining risk.

One of the most effective object recognition features within the 2D domain is Dalal and Triggs' Histogram of Oriented Gradients (HOG) [63]. This feature revolutionized the area by creating a local descriptor that was resistant to both geometric and photometric changes. HOG uses the gradient between pixels to define an angle within a set region,

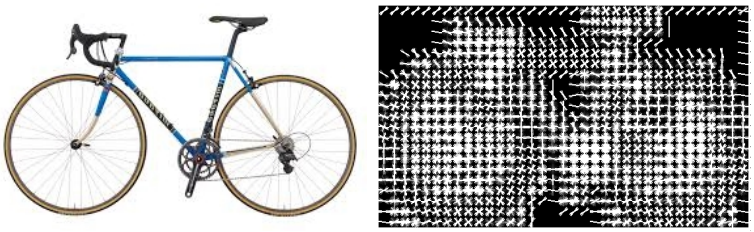


FIGURE 2.5: Example HOG output; left, original image [7]; right, HOG descriptor visualisation.

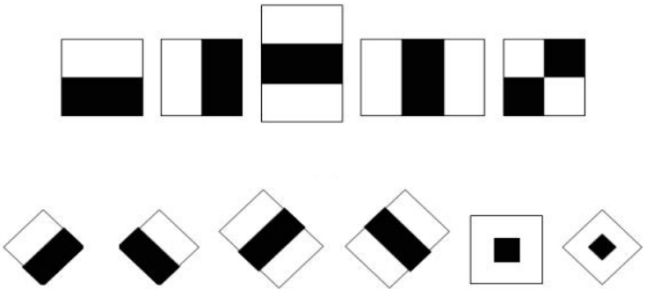


FIGURE 2.6: A set of Haar-like feature examples, used to define a specific region of an image [8].

a group of areas are then put into a histogram containing bins of those gradient angles, when visualised these present a method to see the dominant gradient within an area (Figure 2.5).

Another key feature that is widely used is the Scale Invariant Feature Transform (SIFT) feature proposed by Lowe [64], this method provides a robust feature that was invariant to object changes such as scale, rotation, translation and scene illumination conditions. Sun et al [65] makes use of SIFT in the analysis of large aerial photographs.

The Harr-like feature [66] was used in the first real time face recognition implementations. The feature itself is based on the Haar wavelet principles. In this case an image is classified by regions in which the difference in average intensities are calculated. How a region is divided up defines what types of image properties the Haar-like feature describes well. Using a range of different division formats (Figure 2.6) a final feature vector can be established that represents a broad range of objects or object properties. This vector is then passed to a learning algorithm such as Support Vector Machines (SVM) or Adaboost to produce a robust classifier for a given object. Like SIFT this too is still a very actively used algorithm for modern applications such as for real time road sign detection [8].

Felzenszwalb et al [67] creates a highly accurate object detection method through the use of deformable part models, each of these models represents a part of an object and

is made up using both coarse and high resolution HOG features combined with a spatial model. Part models are robust and when combined allow for detection of highly variable object classes.

The work by Buch et al [68] implements a vehicle recognition framework which can differentiate types of vehicles in real time and provides a method for tracking. This is done using a patch definition system on a 3D representation of the found vehicle, combined with a traditional two dimensional implementation of HOG descriptors to create a robust classifier that is used to identify the vehicles.

With the advent of cheap 3D depth camera hardware, such as the Microsoft Kinect [20], work has been done to transfer many of the well known 2D descriptors into the third dimension. Scherer et al [17] does gradient computation in 3D using a convoluted distance field. This provides an effective way of calculating the magnitudes of the gradients, scoring them highly when localised near a surface of a model (local maxima), however their method also scores highly those at local minima creating additional artifacts within the data.

Another example that uses a variation of vectors within a histogram as a feature is [69]. Here the normal vectors are used as the feature to define an object. An alternative method in which HOG is extended into a third dimension is presented by Klaser [70, 71]. Here a method is proposed which tracks people and identifies their actions through a video sequence. They implement and then extend HOG through use of time as the third dimension. This allows the creation of spatiotemporal features that can be used for action recognition in video sequences. This approach is based on 2D image based intensity gradients but fails to take into account concepts related to the density of an area.

Tombari et al. [72] examine local 3D descriptors and define two main categories in which they fall; signatures and histograms. Signatures are potentially highly descriptive through the use of spatially localized information, whereas histograms sacrifice descriptive power for robustness through compression of geometric structure into bins. The Signature of Histograms of Orientations (SHOT) feature is presented, which encodes histograms of the normals of the points within a neighbourhood as well as introduces geometric information concerning the location of the points within that neighbourhood.

Frome et al. [35] utilise 3D shape and Harmonic shape contexts to build a feature descriptor to find cars in point cloud data. The feature descriptors are defined for an arbitrary set of basis points within the point cloud and are compared using distance measures, such as L2, to a predefined reference set. The methodology is demonstrated on an extensive car database in both the presence of clutter and noise.

Cirujeda et al. [73] presents a descriptor based on the covariance of features, combining shape and color information of 3D surfaces. Multi-scale covariance descriptor (MCOV) has a number of properties including: invariant to spatial rigid transformations, robust to noise and resolution changes and is applicable to characteristic point detection. Additionally, features are defined using a multi-scale framework, which helps link the various features not only on a local scale but also at a more global level too. This has the advantage of reducing repeatability problems and improving detection of points in edges or borders of scene objects.

Rusu et al. [74] proposes an extension to their already well known Point Feature Histograms (PFH) in the form of Fast Point Feature Histograms (FPFH). Point Feature Histograms use multidimensional histograms to capture geometrical properties of a point's k-neighboring points. The use of the multidimensional histograms results in an informative feature representation with benefits such as: invariance to the 6D pose of the underlying surface and handles well different sampling densities or noise levels. FPFH is considerably faster and can be computed online due to a reduction in computational complexity to $O(k)$ (over $O(k^2)$ for PFH) whilst retaining most of the descriptive power of the PFH.

Flint et al. [75] combines the advantages of SIFT descriptor and the SURF detector to produce the Thrift 3D feature detector. Thrift utilises 3D Hessians and creates a weighted histogram of the deviation angles between the normals of points in the neighbourhood of the original feature point.

Finally, the work in [76] uses point pair features to define global model descriptors aiming to recognise similar objects within a point cloud scene. The feature is based on the distance between the point pair, the angles from surface normal to point pair line, and finally the angle between the two normals. Then, using a voting system, it matches pre-defined features to objects in a scene. This system presented good results for object

recognition, but operates on a global scale, making it unsuitable for the concept of local object property recognition.

In many of the given cases above, the features developed are designed to describe specific aspects of the scene, e.g searching for a specific object. For the concept of risk estimation this is a possible approach but relies heavily on comprehensive knowledge of which objects are considered hazardous. A more generic system which looks for the properties of the object themselves, i.e blades, rather than a knife, would allow for a more general and robust system.

Another possible approach to feature description is through the use of a Convolutional Neural Network (CNN) or other deep learning architecture to learn intricate structure of data [77]. Using a multilayered approach and principles of back propagation, a system can be trained to define an optimum combination of representations to perform classifications tasks. Utilising these techniques has seen a dramatic increase in accuracy for many differing tasks including image recognition [78] and speech recognition [79] to name a few. Two primary factors have allowed for these advances, one is the accessibility of large datasets and second is the availability of high performance computing. In the case of risk evaluation, datasets on the scale required for CNN's are not currently available and therefore hinder their use in this area.

2.2.2 Machine Learning

Machine learning is the process by which a computer system analyses data and learns a specific characteristic. Mitchell [80] defines machine learning: 'A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks T, as measured by P, improves with experience E.' From a scene analysis view, machine learning can be implemented to help learn classifiers based on features to define a type of object. This is done by providing a machine learning algorithm with training data in the form of defined positive and negative instances of the classification problem. The resultant classifier will enable new cases to be tested. The use of machine learning techniques coupled with suitable risk specific feature descriptors allows for the creation of a model to identify hazards in new environments.

One factor that must be considered with regards to learning algorithms is the quality of the training data. This must be carefully selected and relevant to the classification problem. Choice of training data has a direct result on how effective the final classifier is. If training samples are too similar the model will have issues of over fitting, too general and the model will not be specific enough to perform the classification task.

Two of the most widely used machine learning algorithms are Adaboost and Support Vector Machines (SVM). Adaboost [33] is a learning technique that creates a non linear classifier to separate data into two groups. Weak classifiers are established using an iterative process with a final strong classifier being a combination of these. At each iteration the weak classifiers with the lowest error margin are used to define the next, this is done in a 'greedy fashion'. Once the algorithm has converged or has reached the maximum number of allowed iterations, the last defined classifier should be the one that best divides the training data.

Support Vector Machines, originally proposed by Vapnik [81] are supervised learning models that are used to classify data into one group or another. This is based on a training set of data that has already been classified into two groups. As SVMs only are able to define a linear classifier this presents problems with datasets that simply cannot be defined by a single plane. To tackle this issue kernels are utilised to map a dataset to a higher dimensional space in which a single linear plane can be defined to separate the data.

An extension of the SVM was proposed by Drucker [82], in which it was put forth that SVM could be used as a regression tool. Regression is the statistical process of analysing data sets to discover a relationship amongst its variables. Often used in the areas of forecasting and data analysis [83, 84], it provides us with the ability to define and explore relationships between dependent and independent variables. Support Vector Machines for Regression (SVR) allow for relationships in future data to be predicted based on a model produced from training data. One of the primary advantages of this model approach is that the required test data can be very small.

2.2.3 Physics Engines

To allow for the definition of an objects stability within a scene, a method is required allowing the prediction of how an object will react under a given input. Physics engines provide this functionality and allow the definition of object behaviour through the application of modelled physics principles. Physics engines play an important role in many fields such as science and engineering [85], entertainment [86], and education [87] allowing the simulation of physical properties of our world in a controlled environment. The ability to simulate physical behaviors is critical, helping us to understand the laws of motion, matter, space and time. This aids in improving design and realism in various industries such as multimedia and game applications, special effects and real-time rendering. In [88] the fundamentals and basic methodologies for physics simulation can be found.

Simulation has two main approaches: high-precision and real-time [89], the first concerns itself with ensuring a simulation is as accurate as possible. Typically employed by industries where the outcome is either not dependant on time or designed to run on high end computer hardware, such as product analysis software or weather simulation. The other end of the scale prioritises a realtime frame rate over accuracy, whilst still trying to emulate as realistically as possible. Examples of industries that utilise visual simulation techniques where speed is a consideration include game production, and real time simulation environments such as flight simulators.

One of the most basic simulation principles is Rigid Body Dynamics; using Newton's laws of motion and the principle that the objects being effected cannot be deformed, realistic movement in a virtual environment can be replicated using simplified computations. Another major aspect of physics engines is their collision detection capabilities. This allows the management of object interactions within a scene based on the concept of simple collision shapes representing complex objects. When any of these objects interact, another action can be triggered, in a simulation world this will often be the application of the resultant forces as a consequence of two physics enabled objects interacting.

Research into physics simulation continues; Baraff put forward a dynamic simulation approach for rigid bodies in [90] and in [91] implementation techniques for real time rigid body simulation were suggested. Physics modeling for computer games development and other multimedia applications was also analysed in [92–94].

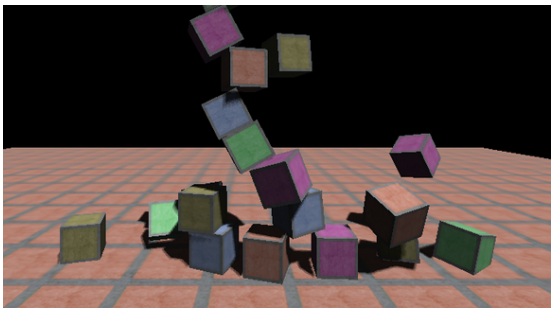


FIGURE 2.7: Boxes falling simulation in 3D environment [9].

Furthermore, simulation provides the means to virtually model a situation and record, at each time frame, the various properties of a scene's objects. Consequently, statistical analysis on the subjects of a simulation can be performed providing further information about a scene's properties. This can help define context or even be used as a feature vector for comparison.

2.3 Human Behaviour Modelling for Risk Evaluation

2.3.1 Simulation Algorithms

To create a more comprehensive picture of risk within a scene, it is important to take into account the way users in an environment interact with it. Although long term observation of an environment can provide this information, it is not a viable course of action when evaluating risk in a new unknown environment. As such the ability to simulate the way humans behave and react to hazards in an environment reduces the time needed to create a useful risk score.

Simulating behaviour virtually has been an area of intense research in recent years although there has been very little focus on the emulation of risk behaviour. The ability to simulate how someone in a given environment is likely to interact with the world around them has a huge range of applications including pedestrian facility suitability and capacity analysis [95], computer graphics and gaming [10], the social sciences [96] and engineering [97].

In general, two method types are used to model pedestrian flow. The first, usually applied to large crowds, involves treating the agents as a whole, usually as a fluid or

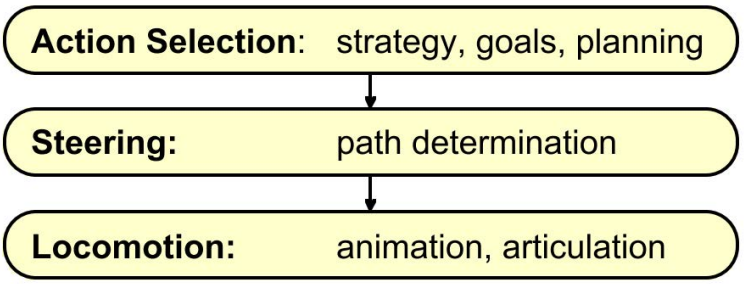


FIGURE 2.8: The three layer simulation hierarchy as defined by Reynolds [10].

continuum which responds to local influences. These types of methods tend to be referred to as Macroscopic [98, 99]. Macroscopic methods present a number of advantages, the computational requirements to process large scale crowds is lower and the effects of unforseen behaviour of individual agents has less impact on the crowd overall. The fundamental issue with this approach is the tendency to model the direction of movement and speed of a pedestrian as a single flow-density relationship with those around them. This does not easily upscale to more complex problems, including multidirectional simulations, nor does it take into account the motivations of the individual pedestrians as they are all represented identically. For risk behaviour analysis this is not viable as it is this agents individual reaction to risk which will effect their movement as well as those around.

The second method, microscopic, treats pedestrians as discrete individuals in a simulation. This allows a better level of granularity on the motivations behind an agents direction, speed and position. This provides the functionality to allow decision making properties and risk related variables to be considered for agents individually. Due to the individual nature of each agent, these methods tend to have a higher programming overhead and computational cost. However as the cost of computational power is always going down, the microscopic models are tending to be favoured for their ability to better emulate human behaviour.

Simulating behaviour is often considered to have started with Reynolds work in 1987 with the emulation of animal behaviour [100]. Within this work the concept of steering simulation was introduced, whereby each individual agent in a scene has its movement governed by a set of rules. In this case each agent follows three rules: steer towards the goal, steer away from the nearest obstacle, and steer away from the nearest person. When obstacles are very close by or when collisions are imminent, the avoidance rules are given precedence over the goal-following behavior. This was further developed by Reynolds

[10], here the steering behaviours model was better defined for gaming applications as a three layer hierarchy (Figure 2.8).

Helbing et al [101] introduces the Social Force Model (SFM) which uses potential fields defined by neighboring agents to impart an acceleration to each agent. SFM's compute the trajectory of each agent by applying a series of forces to each agent that depend on the relative positions and velocities of nearby agents and the goal of the agent. As an example an agent will have a goal which they are progressing towards at their set speed. If that agent is in a crowd, a repulsive force, pushing it away from each neighbor, will be applied to maintain a personal space. Additionally a force pushing it away from walls or obstacles would also be applied. The magnitude of these forces decreases exponentially subject to the distance from the agent. One observed effect of the social force model is the emergence of behaviours within the agents present in crowds, such as line forming in tight areas. Another key advantage of this model was the use of variables that related to physical principles in our world. The use of these parameters allowed the application of other forms of research to drive the simulation and formed a basis for evaluation.

Another example of this type of approach is proposed by Xi et al [97], in which a dense model is proposed based on the inclusion of a number of decision making factors to each agent. A model integrating extended decision field theory for tactical level human decision making, the social force model to represent physical interactions and congestions among people and the environment and a dynamic planning algorithm involving AND/OR graphs. Extensive testing is done on potential profit for a shopping mall when various factors of an agents AI are changed. For example experimentation with group dynamics or visual length. Additionally variables effecting likelihood for a shopper to browse or intention to buy are also changed and reported. However no real validation based on real results is presented. Survey and observation data is used in the setting of these model parameters.

These outline two popular models of automation for virtual agents, however they are somewhat reactive in the way they work. High level strategy components can be implemented, such as A* path planning [102], to avoid static obstacles within a simulation. However for dynamic entities, an agent will have to get within a certain distance of an agent before an applicable force will have sufficient impact for their course to change. This can also lead to unrealistic or unnatural movement behaviour in cases of agent

avoidance. Predictive planning based simulators attempt to anticipate collisions based on neighboring agents' positions and velocities and determine new paths which avoid these collisions.

Karamouzas et al [103] use a predictive collision avoidance model which focuses on modelling a humans ability to predict future collisions and take a minimal avoiding action. For instance, rather than changing direction, slowing down slightly may reduce the impact to the overall journey time. By calculating the future position of an agent at a forthcoming time step, collisions can be detected and an avoidance force can be applied. This avoidance force is calculated by analysing the vector between the future points.

Zheng et al [11, 16] use a rudimentary form of behaviour modelling for their human interaction contribution to risk. Using an evaluation of the average movement of a human captured using kinect skeletal data, a local disturbance field for a human in an environment is created. This is extended with a second disturbance field constructed of all the possible paths a human might make through an environment creating a global disturbance field for the scene.

These methodologies define a number of interaction and psychological principles that dictate how humans traverse environments. This is useful when determining human interaction with a scene under normal circumstances, however in the presence of risks that behaviour might change. As such it is important to take into account these changes in behaviour when modelling human movement to ensure that a more accurate representation is given under more complex scenarios, such as in the presence of risk.

Stroeve et al [104] looks to model the situational awareness (SA) of airport workers in the context of human performance modelling in accident risk assessment. Endsley [105] defines SA as follows: situation awareness is the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future. Using this concept a mathematical model is constructed for the airport scenario, using various different agents (such as planes, workers, drivers) and a scheduled SA update which refreshes an agent's knowledge of the area. Risk calculations are made based on a number of risk scenarios, such as crossing the runway. Specific simulations are not run, however using the constructed model,

scenarios are created that highlight which aspects of airport safety are most relevant to worker safety.

Kim et al [106] adds a stressor component into their simulation algorithm. The addition of this element allows for agents in a scenario to disregard common psychological behaviour effects that maintain coherent movement such as separation from those around. Instead the agent focuses on reaching their perceived goal as fast as possible. Through this addition the ability to model instances such as building evacuations under stress become more accurate. As a test case, the scenario of the Shibuya Crossing in Tokyo is utilised. A quantitative assessment is made of some of its modelled scenarios by looking at psychological studies of specific situations and looking for similarities in the simulation. For example, in a study of pedestrians crossing roads [107], average crossing speeds were measured against how much a pedestrian has been delayed entering a street crossing from the start of the signal. This is then compared with the outputs of the simulation to provide a similarity measure.

Klugl et al [96] creates a model that aims to simulate an agents response to the evacuation of a train with an engine on fire in a tunnel. Human behaviour is simulated such that an agent is able to take into account information such as perceived heat and smoke levels, as well as input from other agents pertaining to the source of the fire. Exits are given as either the end of the tunnel or specific emergency exits. Particular attention is given to the concept of an agent's spacial requirements, as the motivation for the project was finding a balance between evacuation efficiency and build cost of the tunnel. This aims to provide insight into human behaviour in emergency situations. However validation of this type of behaviour is difficult due to lack of available data.

2.3.2 Metrics for Simulation Accuracy

With the advent of so much research in the area of pedestrian simulation the ability to determine what is a good simulation or an accurate simulation has also been a well researched topic. Simulation accuracy is crucial to ensure that the outputs produced are a good representation of human behaviour. In the case of hazard evaluation this is especially important as the safety of human subjects may be at risk. Due to the large array of applications that these simulations have, what constitutes a better simulation is very much a topic for debate. Currently evaluation tends to be split into qualitative

and quantitative. Where the former will seek to judge a simulation using experts in the relevant field or with category based rating systems in which aspects of a simulation may be graded, quantitative measures seek to provide a numeric measure of accuracy for a simulation. These evaluation techniques tend to be data driven, and as such tend to require some kind of ground truth data from which to test against. Here the evaluation is based on how similar the algorithm used replicated the pedestrians in the source data. Both methods have advantages in certain situations, however they are not interchangeable and are therefore not comparable.

For example, Klugl's [96] work on train tunnel evacuation simulation evacuation. Here the most important factor is if the time it takes the simulated crowd to exit the environment is similar to that of a real crowd. Evaluation of this particular problem is challenging as accurate recorded data of these types of events is limited or not readily accessible. As such a direct data driven comparison of an agent's behaviour to that of a recorded pedestrian is not possible. Instead evaluation is conducted by those who have experience of the situation. Looking for behaviours that are missing from the simulation or those that are unnatural.

The primary issue with these types of evaluation is that a context independent form of comparison does not exist. Evaluation for a simulation algorithm is targeted at the field it was developed for. To this end many surveys have been done on existing simulation approaches by putting them through the same test environments and evaluating which perform better. A large survey of 27 simulation models has been done by Duives [108], covering both micro and macroscopic models. Each model is rated in its performance across 24 test criteria in three main fields. These include categories like computational performance, the presence of emergent behaviours as well as the presence of particular model abilities such as route choice and agent strategy. This extensive survey goes a long way in the definition of comparative attributes that a simulation can be tested against, however the rating system applied to the criteria is not specific and no quantitative results are defined.

Similarly a survey of pedestrian behaviour models has been done in [109] which focuses on assessing simulation approaches considering various aspects of pedestrian behaviour in urban environments. Here focus is mainly on the concept of analysing pedestrian models in relation to traffic crossings and the associated decision making processes needed to

accurately simulate these behaviours. There is extensive discussion in general terms about the effectiveness and drawbacks of each approach, however no clear definition of metrics is provided.

Both survey papers [108, 109], provide a good demonstration of the types of qualitative evaluation techniques often used. They evaluate the effectiveness of a methodology in specific situations, such as presence of specific behaviours or features, but fail to draw any strict conclusions about which methodology performs better or worse. Additionally there is no real analysis of how the simulation 'looks', i.e if the methodology produces visually similar or natural behaviour, which is often a key requirement of the design.

In more individual cases it is seen that often the evaluation technique is designed to fit the context. This of course makes sense but often means that other key aspects of a simulation implementation are not analysed. Portez [110] focuses on the simulation of crowds around bottle necks and looks for specific events. Here density matching against recorded video footage is used as a quantitative measurement, specifically the number of people per square meter. This is backed up using visual checks against the original footage to ensure the simulated crowds resemble those in the captured data.

Asano [95, 111] focuses on the concept of pedestrian collision avoidance at a local level with an additional high level tactical model. This aims to have the pedestrians avoid areas of high congestion. The local level model uses an anticipation period in which an agent can predict if a collision with another agent or obstacle will occur. Collisions are considered likely, based on the prediction of the agents trajectories crossing. In such cases the first agent to get to the crossing area will have priority, with the other 'giving way' by reducing speed. Assano uses their own captured ground truth data to evaluate their model. When implementing their algorithm on a larger simulated environment, crowd density data captured from a real scenario is used as the ground truth to test against. Both methods provide a strong grounding from which to evaluate their methodology quantitatively, however the need to define accurate ground truth data introduces a number of data collection issues. For example, the crowd density analysis tool that was used provides only an effectiveness rating of 80%. This starts to introduce wide margins of error into the evaluation process that could be avoided.

In other approaches, in order to evaluate the simulation accuracy, ground truth is obtained using mobile device tracking techniques [112]. In these cases obtaining the ground

truth can have issues pertaining to cost, requirement of specialized equipment, ethical and privacy restrictions, suitability for environment and can be time consuming. Obtaining ground truth data for risk evaluation poses further problems, such as the unexpected nature of the events making controlling the environment difficult.

Lerner et al [113, 114] address the concept of look and feel of a crowd by assigning a similarity metric to individual agents in a scene by comparing their actions at a given moment in time to a database of observed actions. The constructed example database is taken from recorded videos of both sparse and dense crowds. These have been manually annotated to record individual path vectors for every person in the footage at each frame. A state-action pair for that frame is defined using firstly a state (set of recorded variables such as trajectory, speed and position) and an action (a density measure). The density measure analyses, for a given frame, the number of people in a set of defined regions around the subject agent for a two second period, providing a compact representation of local density changes over time. The similarity between a state-action pair from the database and a test state-action pair is defined as the similarity between the actions (differences in positions along the trajectories) and the distance between the states (differences in densities for the surrounding regions).

The fundamental issue for this method is the requirement of ground truth data and the associated issues with its capture. When trying to evaluate a simulation algorithm that emulates the presence of risk this type of data is difficult to obtain. However retrospective positional information can be obtained from video footage, for example CCTV. This falls into the area of pedestrian tracking and has also seen much work recently [115, 116]. Specifically, techniques have been developed for estimating the flux of people in public areas, such as stores or travel sites, which can then automatically provide congestion analysis assisting in management of crowds and pedestrians [117–119]. Use of more specific analysis techniques such as tracklets which allow the track of a specific pixel or area of pixels though a scene can also be used to generate relative positional data. Another applicable vision technique is the use of optical flow. The majority of today's optical flow methods strongly resemble the original formulation provided by Horn and Schunck [120] as well as the work by Lucas and Kanade [121]. The accuracy and robustness of optical flow estimation algorithms has seen significant improvement over the last decade [122, 123]. A technique that incorporates optical flow for accuracy evaluation in crowd simulation was proposed in [124]. In this work a solution is proposed

which allows the relationship of optical flow to physical velocity to be defined. However it requires manual annotation and performs well only in specific relative orientations of the camera and pedestrians.

2.3.2.1 Simulation Evaluation Frameworks

Overall the existing simulation accuracy evaluation techniques provide partial solutions to the issue, often the concept of human behaviour outside of normal situations is not taken into account, for behaviour pertaining to risk this presents an issue in validation. Due to the size and potential applications of a global evaluation technique, better dedicated tools are required. Simulation Evaluation Frameworks provide another form of simulation evaluation which better facilitate the methods of comparison.

Charalambous et al [125] looks at the creation of a user-in-the-loop analysis tool that takes a simulated environment and, by comparing it to reference data, characterises outlying behaviour. This can take the form of unusual paths taken or abnormal levels of speed. Two processes are suggested, outlier detection which takes a set of data and searches for outliers, which allows the definition of odd behaviour within that dataset, however this would not pick up systemic issues with the simulation. Secondly novelty detection, in which sample data is compared to reference material to find and describe trends or actions that differ from the reference data. Finally the results of the analysis are presented to the user in a number of forms that aim to highlight specific agents that are acting erroneously or where general areas of inconsistency appear.

The fundamental issue with the process is the inability for the method to handle the erratic behaviour caused by unexpected events, hazardous or otherwise. As the process depends on trend analysis, outlier behaviour such as that caused by an unexpected event would be categorised as abnormal. Another issue pertains to the need for the reference data to be very similar to the simulated, indeed both processes required predefined tracks for each agent. Additionally as the comparison made is purely a data driven approach, the concept of visual similarity is not addressed, just because the agents' paths in a crowd differ from the real world examples does not mean the simulation does not *look* real. Additionally analysis is given on an agent level with no global similarity measure given.

Guy et al [126] uses a computed entropy score to compare simulated data to captured real world data. The metric is defined as the entropy of the distribution of errors between the evolution of a crowd predicted by a simulator and the source data. The consideration is made that the source data is subject to noise produced by the capture mechanism. Using a expectation–maximization (EM) algorithm to iteratively calculate this noise, the source data is then corrected allowing for a better quality comparison. Using three differing datasets ranging from simplistic two person interactions to dense crowds, evaluations are done to produce simulations that closely resemble the source data. This statistical analysis relies on the need for position information from both the simulation and comparable real world examples. Additionally a number of assumptions are made, the most noticeable being that the crowd simulator is not systemically more accurate for some agents within a crowd than for others. This is not always accurate as there will always be aspects of a simulation that are more accurate than others. The work in [125] demonstrates this.

Kapadia et al [127] look into the assessment of simulation algorithms on a more global scale. As such they present two notions: the first is the concept of scenario spaces, and secondly, metrics to quantify the coverage and the quality of a simulation algorithm in this space. Scenario space is defined as a set of parameters from which test environments can be generated to test a simulation algorithm in. These values consist of numbers of agents, obstacles and environment size amongst others. A successful space is one in which a simulation agent gets from it's origin to it's goal and does so without colliding with any obstacles or other present agents. This process is repeated for a representative number of possible combinations derived from the scenario space. Evaluation of the successful spaces is done based on three metrics: scenario completion, length of time and distance travelled. These are compared to optimal values calculated when generating that specific test space. Using these metrics, concepts of coverage, quality and failure set are computed for a given simulation algorithm for a given scenario space. This exhaustive form of measurement is comprehensive and effective for testing steering behaviours for the given space parameters. In addition the automated nature of the tests provides a useful platform from which benchmarks can be done. However the parameters for the scenario space have limitations that seem to create a specific type of test which, from the provided images, do not resemble likely everyday scenarios. However due to the exhaustive nature of the tests, some may include more realistic examples. This suggests

that the proposed methodology is not suitable for evaluating how real a simulation can look rather how robust an algorithm is.

Rodriguez et al [128] develops a set of video comparison features for use in their crowd tracking work. Recently crowd tracking has benefitted from the use of global crowd analysis tools to help in the tracking of individuals within the same crowd. Rather than using the same source to perform global crowd analysis, the use of a pre existing database of defined crowd analytics is suggested. This uses a two part matching scheme, utilising a high level refinement step based on a broad Gist scene descriptor, then a 3DHOG representation to closely match examples in the database to the test sample.

Pettre et al [129] introduces an agent interaction method and mechanisms to evaluate it. This uses density plots based on aspects of the simulation reaction times, as well as a Maximum Likelihood Estimation (M.L.E.) technique to define a likelihood function based on a set of proposed simulation variables and the assessed difference from captured data. These provide a good benchmarking tool, however testing is conducted only on two person interaction scenarios and their validity as metrics for larger scale crowd simulation remains untested.

Wang et al [130] present the Stochastic Variational Dual Hierarchical Dirichlet Process (SV-DHDP) model in which groups of similar trajectories (trending paths) can be combined to generate an overall path pattern which consists of flows of location-orientation pairs. The path patterns created are therefore the result of local dynamics and global factors relative to the scene, allowing differing insights based on the simulation environment. The resultant visualisations allow for detailed qualitative analysis and the introduction of an inference based similarity metric allows for the comparison of extracted path patterns from differing data sources. This provides a good generalised view of a subject scene. However analysis is done on defined paths for source and test data which requires complex post processing techniques or data captured in a specific format. Often this type of captured data can have extensive inaccuracies as demonstrated in [95].

Musse et al [131] also address the issue of tracking generalised paths in crowds using four dimensional histograms to describe movement within a crowd. An additional clustering process is applied to identify differing areas of flow within the crowd. By applying the Bhattacharyya distance as a form of measurement between the defined features, similarity is assessed on criteria such as speed, spacial occupancy, and orientation. The

results produce similarity measurements for aspects of orientation and speed but fail to take into account the density of the crowds. Additionally no analysis of what is visually similar is given.

Jablonski et al. [132] use a novel combination of scene reconstruction and composition techniques to compare simulation algorithms against source video footage. This provides a flexible method which can use un-annotated footage of pedestrians. A 3D representation of the environment in the source footage is created and simulation algorithms are used to control agents within that scene. A new video is constructed from a viewpoint similar to that in the source footage. Using these two video samples, comparison is made and metrics used to define a similarity between them, with simulations that emulate the pedestrian actions well, resulting in higher levels of similarity. This use of un-annotated footage is a key advantage when considering risk applications, however the use of fully reconstructed scenes is a time consuming method which could be improved.

The use of comparison frameworks is vital given the need to validate the accuracy of simulation outputs. Risk related simulation algorithms pose a much more difficult problem in evaluation due to the lack of source data and the potentially erratic nature of behaviour. As such a tool that can utilise the sparse available footage would make a valuable addition to the research area.

2.4 Datasets

Within the following chapters a number of datasets are utilised to test the proposed methodologies. An outline of these is given below.

Dedicated risk datasets are not common in the area of scene analysis. In general 3D scene analysis datasets in the area are designed to be broad to allow application in many areas. As such for those works that have implemented risk evaluation in one form or another, datasets have been put together for that task specifically. Zheng et al. [16] for example uses a 120 scene dataset captured for the task, using various depth capture devices and SLAM techniques (Figure 2.9). In these scenes, rooms are captured with a number of objects which are then used in the analysis of fall potential. An extension to this work is presented [11] in which the NYU Dataset [133] is used and analysed in the same way.

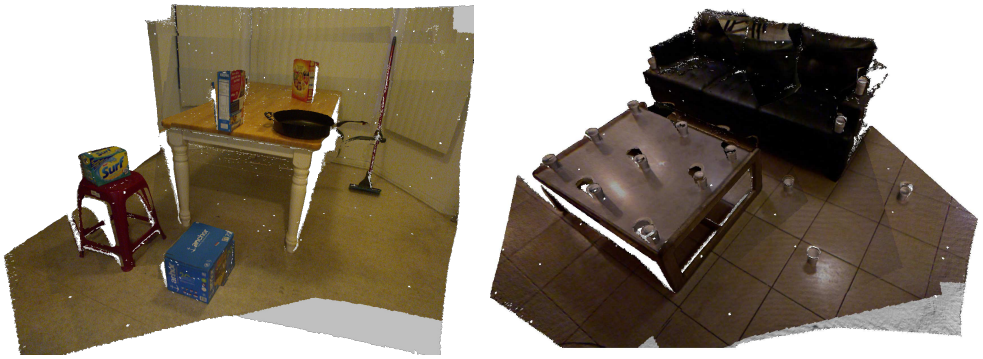


FIGURE 2.9: Example point clouds utilised in the work of Zheng et al. [11].

FIGURE 2.10: Subset of the objects in the 3D Risk Scenes (3DRS) dataset.

To help aid the area of risk related research, specifically in domestic environments, the 3D Risk Scenes (3DRS) dataset is introduced [29]. The goal of this dataset was the inclusion of a range of differing risk related scenes that could be used to evaluate various aspects of potential risk in a domestic environment. As such the dataset includes 3D models of individual objects as well as example scenes of multiple objects on a table. All objects are deemed to be commonly found in a domestic setting including kitchen implements such as cutlery and knives, as well at crockery. Other household objects such as tools, toys and appliances are also included (Figure 2.10). Of the 27 objects captured, 12 are classified as hazardous with the remaining 15 safe and the range of objects is intended to allow the detection and classification of hazardous objects within a scene. Each object is scanned using the Microsoft Kinect RGB-D camera, and utilising 3D real-time scanning software to capture and produce a 3D model of the object. The scan space is an estimated $50cm^3$, proving a challenging level of object detail whilst allowing a large range of objects sizes. In total 27 household objects are contained within the 3DRS dataset with a further 40 synthetic object models created using CAD modelling software. For the work contained in this thesis only those models captured using the RGB-D camera are utilised.

FIGURE 2.11: Some scenes of the new 3D Risk Scenes (3DRS) dataset with the three levels of stability for each one.

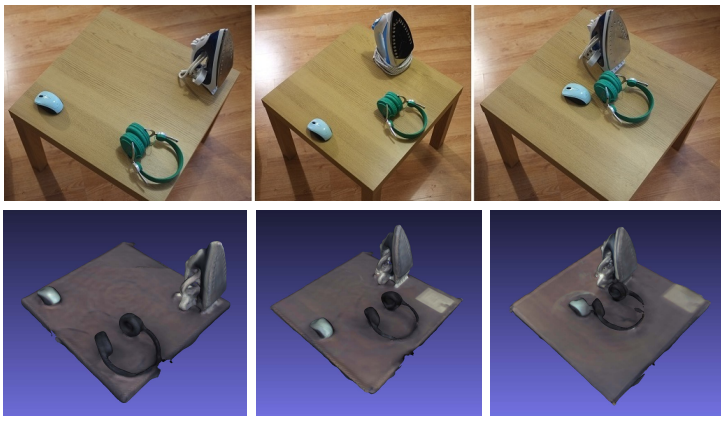


FIGURE 2.12: A scene from the new 3DRS dataset reconstructed using Kinect Fusion for the three levels of stability.

In addition to the individual object scans, 48 indoor risk scenes (Figure 2.11) were captured. Scenes are constructed using a subset of the 3DRS datset's 27 household objects (Figure 3.11). Each scene is configured on a square table and consists of three objects. The 48 scenes are split into 16 of these three object combinations, each having three iterations. For each iteration the objects ion the scene are placed closer together and towards the centre of the table (Figure 3.10). This arrangement of objects is designed to allow the analysis of object stability and position.

Another popular area of computer vision research is facial expression recognition. Many 3D feature descriptors have been applied to this problem and one of the most popular datasets within the field is the BU-3DFE dataset [12]. This dataset presents a multi-class expression recognition task, whereby the goal is to define classifiers that can evaluate the emotion a face is displaying. Within the dataset 100 participants (44 male and 56 female) with a range of ages and ethnic/racial ancestries have their faces scanned and photographed whilst being asked to perform seven different expressions. Each expression

FIGURE 2.13: Example images of participant expressions at highest intensity (happiness, disgust, anger, surprise, fear and sadness) from BU-3DFE dataset [12].

FIGURE 2.14: Example frames of video from the PETS dataset [13].

is performed with four levels of intensity (Figure 2.13). Due to the popularity of the dataset it has become somewhat of a benchmark in this area of research providing an excellent form of evaluation for 3D feature descriptors.

Pedestrian simulation has seen a large amount of research over the past ten years. As a result of this, many datasets have been collected and publicly released for others to use. The Performance Evaluation of Tracking Surveillance (PETS) is a workshop designed to address the problem of crowd image analysis (Figure 2.14). A comprehensive three scenario database was built [13] and ground truth established for a number of assessment tasks. The dataset contains multi sensor views of each scenario. Although this dataset provides a number of manufactured crowd scenarios, the controlled nature of the environment allows testing of methodologies on simplified crowd examples.

The Mall dataset [134, 135], contains ground truth data for over 60,000 pedestrians across 2000 frames of video. The dataset is collected from a publicly accessible webcam and is intended for crowd counting and profiling research. This is a challenging dataset

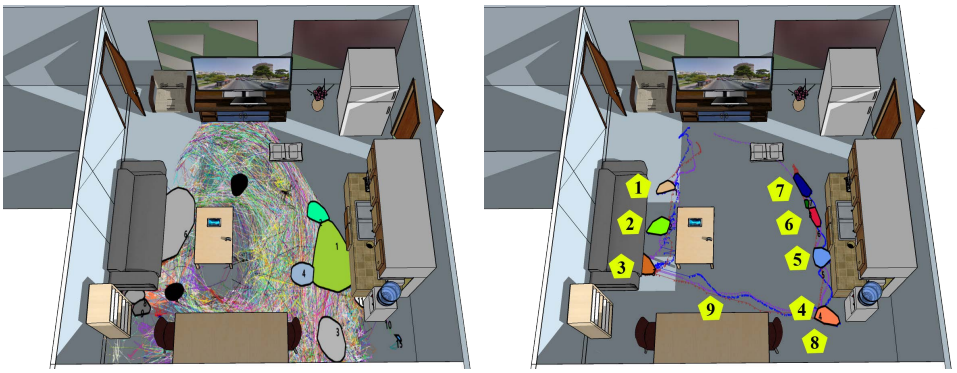


FIGURE 2.15: Long term observations of a sample room and the results of the clustering algorithm used [14**?**].

due to the low frame rate ($< 2Hz$), however it is an entirely natural scene providing a good dataset from which to try and mimic natural movement.

The RBK dataset [136] is another CCTV based crowd and pedestrian dataset. The data is captured from a number of locations around a town centre in the UK. The camera locations range from three to ten meters above ground and are of PAL quality (576 lines of interlaced video), with a frame rate of $24fps$. This dataset again forms a good natural view of more complex crowd behaviour. Scenes involving multidirectional flows and changes in direction also contribute to making the dataset more challenging.

The final pedestrian dataset is a long term observation dataset [14**?**]. Here elderly adults who reside in an assisted living complex are monitored long term though the use of depth cameras. The skeleton information for people in a single room of the house is recorded all day for a period of 12 months. Due to the huge amount of captured data, a clustering algorithm is utilised to monitor areas of interest within the room on a daily and monthly basis (Figure 2.15).

Chapter 3

Stability Estimation for Risk using Physics Simulation and Prediction Techniques

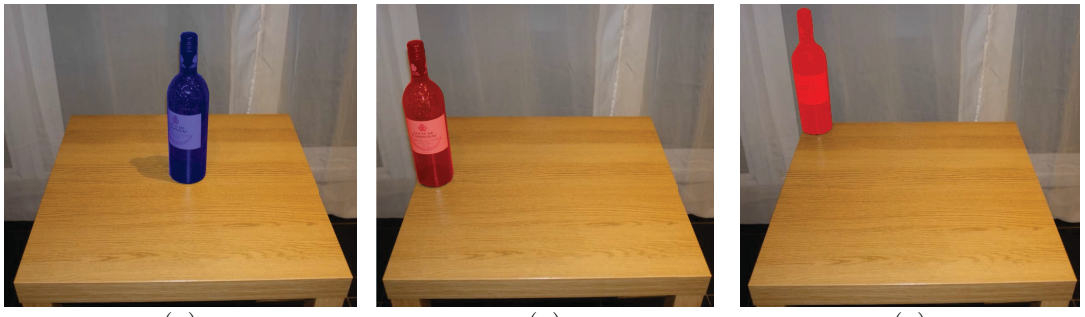
3.1 Introduction

As discussed in Chapter 1, automated risk assessment is a problem that has not been fully addressed. As such, the Risk Estimation framework is proposed which provides a basis from which the combination of measurable elements of risk can be used to output a quantified risk score for a given environment. The framework, and its associated measures of risk, are designed for indoor applications, specifically domestic environments. Due to the intended area of application, the focus is given to risk detection for situations that are pertinent to the home setting, i.e the detection of sharp objects commonly found in a home or the stability of those objects within the environment.

One of the primary issues with a unified approach to risk estimation is the problem of context, ensuring that a provided risk score is relevant to the end user regardless of situation. What can be considered safe in one environment may not be in others. For example, a container of liquid at the edge of a table is risky in a household environment however in a chemical laboratory this might pose a far larger danger. In the case of the suggested measures of risk in Chapter 1, stability would need to take precedent over hazard features or human behaviour analysis. Similarly users of the environment will

also affect how risk is perceived; if the environment contains children or elderly adults the threshold of what is risky may need to change. However, regardless of context, the elements that might contribute to the concept of risk can be broken down into components from which a decision can be made. These components include elements such as shape, size, material, temperature, position and many others. Using the output of the Risk Estimation framework, domestic robots could be trained to help avoid potentially hazardous situations. In the smart home example attention could be drawn to these situations and thus accidents avoided.

Within this chapter the first element of measurable risk will be proposed, in the form of stability estimation for objects in a scene. Going forward the term stability estimation pertains to a quantifiable measure of an object's instability within its environment, such that objects placed in more unstable positions within a scene are given a higher score. Using this analysis of stability, it is possible to gain an understanding of whether the placement of an object in a scene presents a potential risk. In a basic example, the returned risk score from the Risk Estimation framework for a glass bottle at the corner of a table would be higher (more unstable) than one placed at the table's centre (Figure 3.1).



(A) (B) (C)

FIGURE 3.1: Bottle on a table with three levels of stability. (A) Centre of the table, most stable. (B) Edge of the table, more unstable. (C) Corner of the table, most unstable.

To analyse how unstable an object is within a scene, a physics engine is used. Simulating the application of forces to objects within the environment and monitoring the outcome allows a picture to be built up of instability for a set of objects. The advantage of the physics engine approach is that an output of object position and rotation is given per frame of the simulation. Using this information and measured kinetic energy, an output can be created based on the amount of energy produced by an object, as a result of an applied force. As such with the application of forces from a sample of directions around

an object, an instability diagram can be produced representing areas and objects of high or low instability.

The main advantage of using a physics engine as opposed to other techniques such as probabilistic models [16] for each object, is the ability to simulate collisions with additional objects and therefore allow the propagation of the instability within the scene. This allows the holistic analysis of a scene when measuring the stability of each object. As a result, outputs to the Risk Estimation framework can be in a number of forms; per object instability or instability of the scene as a whole. This helps to further tailor the Risk Estimation framework for the required context, as a global definition of risk may not always be appropriate.

There have been major advances in the development of physics engines in recent years. The increased use of physics engines in other research areas and applications such as robotics and engineering, medicine, and training applications [85, 137, 138] has led to the development of faster and more precise engines. In tandem the increase in computer performance has allowed and facilitated the development of such engines.

Physics engines tend to focus on two main areas when it comes to simulating the physical world. The first is concerned with high precision, aiming to replicate the physical world as accurately as possible. The second is focused on the speed at which it can calculate these physical representations.

When computational power is restricted, compromises must be made with regard to the accuracy of replicated simulations. This is often in the form of reducing the number of objects that are being simulated, simplifying the scene or sacrificing the simulations accuracy. Due to the ever increasing physical complexity of scenes, the need for real-time rendering and limitations of computer hardware, it is important to find a balance between accurate representation and calculation time. The balance being based on a set of parameters such as relevance or type of simulation. The estimation of an object's stability in a scene would be a costly process. This is in part due to the need for a large number of simulations per object in order to build up a general picture of an object's stability within the scene. With this in mind a method of reducing the computational requirement of this process is required.

As such, within this chapter the following contributions are presented: The Risk Estimation framework is described and defined, outlining the ability to take as input elements of measurable risk. Additionally through the use of weighting, context is taken into account providing risk scores that are relevant to the end user regardless of environment. The first element of measurable risk is also presented; a novel stability estimation method using physics simulation which is evaluated against the current state of the art method. Finally an approach to physics simulation using machine learning and dimensionality reduction is presented. Through the use of a prediction framework the computational requirements needed for such processes are reduced. The proposed framework learns complex physics scenarios for an environment and, during a simulation, dynamically switches to a prediction mode for a certain amount of time thus reducing the overall complexity and computational workload for a given simulation.

3.2 Related Work

The concept of defining the stability of an object has been raised in the research before, however research relating to risk applications is sparse. Zheng et al [11, 16], evaluate risk in a scene through the analysis of the probability that an object could be dislodged through the use of disturbance fields. Human interaction as well as natural disturbance is modeled to create risk scores for objects within the scene. Using this data a potential falling hazard is calculated as a function of an object, the applied disturbance and the amount of energy required to dislodge the object. This highlights objects in a scene that are in a position where they are likely to be dislodged. This use of a probabilistic model to identify these objects is efficient as it avoids the need for costly physics simulations. However the result is simply an indication that this object may be in a hazardous position. It cannot tell where the object will fall or from which angle the force needs to be applied, nor does it take into account other objects in a scene. Additionally the risk score is based on a specific type of input, which requires modeling per event.

Other stability based techniques have been used to further enhance other aspects of scene analysis. Wu et al. [52] makes use of physics engines to aid in the process of learning. Their work aims to emulate the idea that a human's ability to analyse a scene

is based upon a realistic physics engine as part of a generative model to interpret real-world physical scenes. The result of the defined system is the ability to output physical properties of objects from video observations such as mass and friction coefficients.

3.3 Methodology

3.3.1 Proposed Risk Estimation Framework

The definition of a unifying framework that can produce a risk score from any measurable element of risk forms the basis of this proposed work. As such the cumulative risk score R for a scene is defined as the weighted sum of n measured risk elements e (3.1). The weighting specified for each element should fall into a range from zero to one, with the sum of the weightings for all included elements being equal to one.

$$R = \sum_{i=1}^{n} (w_i e_i) \tag{3.1}$$

A risk element is any quantitative measure that could highlight potential risk. These elements could include concepts such as stability, hazard shape features or any other properties that may present a danger, for example temperature obtained from a thermal camera or material analysis data. Each of these elements has an assigned weight; this allows the context in which the risk is being evaluated to be considered, applying more weighting to elements that are more relevant in a given situation. For example, in an environment with adults present, stability may not have a weighting as high as in situations where children are present.

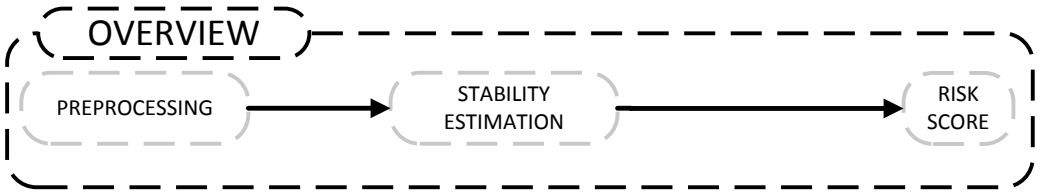


FIGURE 3.2: Overview of the Risk Estimation framework with the stability estimation element.

For the purpose of this work we define the risk score R as a function of the weighted element of stability S for the objects within the scene. Figure 3.2 outlines the overview

of the Risk Estimation framework from preprocessing, in which the scene is captured and processed to a usable format, through the stability estimation process and then finally the resultant risk score.

$$R = w_S S \tag{3.2}$$

3.3.2 Preprocessing

Before a scene can be analysed for its stability it must first be digitized and preprocessed into a suitable format. Figure 3.3 demonstrates this process.

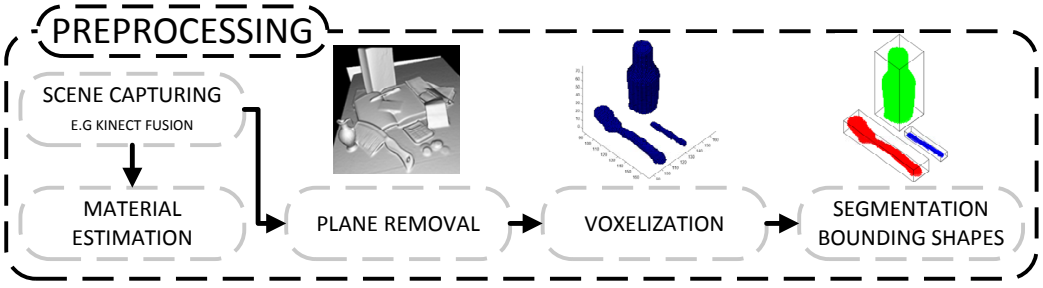


FIGURE 3.3: Preprocessing steps: Scene capturing with Kinect Fusion or similar SLAM technique, plane removal, voxelization and segmentation.

The target scene must be digitized for further preprocessing to take place, a 3D mesh model reconstruction is created using methods such as Simultaneous Mapping and Localisation (SLAM) techniques e.g Kinect Fusion [4] or multi camera acquisition systems [39]. Other sensors such as thermal or acoustic cameras could also be used. Each of these methods return a three dimensional representation of the subject scene. These can be either: a point cloud, a set of individually defined points in 3D space with additional colour information, a 3D model, built up of points in 3D space (vertices) which are connected by edge segments to form a polygon mesh or finally, in a voxelized form (described in detail below). In this work, scenes have been captured using Kinect Fusion, using a Kinect camera which returns a point cloud.

The surface on which the objects are set requires removal prior to segmentation. In the case of the given scene this represents the table surface on which the objects in a scene are set. The work by [41] presents a solution to this using connected component based clustering within a point cloud together with a 'planar refinement step'. The dimensions

of the removed plane are recorded and used later to define the surface during simulation. Once the plane has been removed the point cloud is converted to a 3D mesh model.

The returned 3D model then requires conversion to a data format that is suitable for use in the provided methodology. Voxelization is used to produce an equally spaced grid representation of the scene, where each voxel provides a binary classification of either object or not. For this process existing techniques are implemented based on the work in [40]. Initially a grid is defined in 3D space around the model. Each cell of this grid, in 3D, represents one voxel. Using the vertices of the model and a defined radius, voxels with a centre which falls into this area are defined as part of the model. Using the model's edge information a cylinder is defined along the length of the edge, voxels with a centre which falls into this cylinder area are then classified as part of the model. Finally for a given face of the model, two additional planes are defined above and below the surface of the given face and all voxels with a centre which lies within this area are attributed to the voxel representation of the model. At each stage of this process rules are applied which maintain a hole free voxel surface. The rules define relationships to neighbouring voxels based on the model data. The voxel representation is optimised based on principles of accuracy, minimality and separability, where accuracy is defined as a measure to quantify how well represented the model is. Separability, which could be described as the appropriate separation of voxel space using the defined voxel surface. Finally minimality, which ensures that additional voxels are removed subject to accuracy and separability. This step may be avoided if the data capture method returns a voxelized representation of the scene [139].

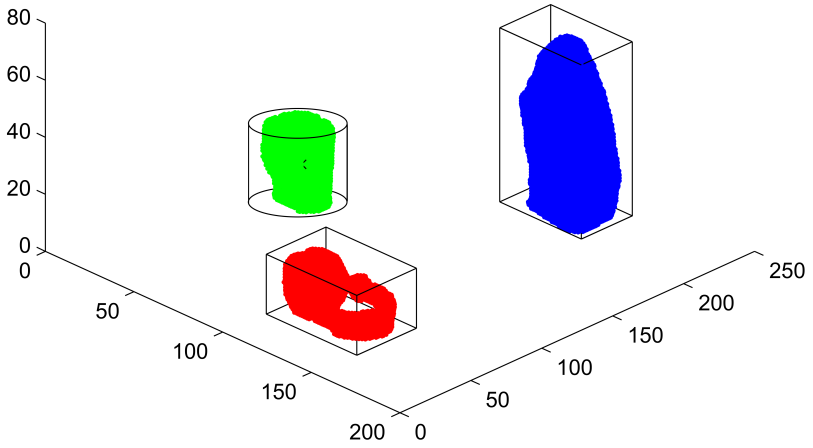


FIGURE 3.4: Example ideal scenario, captured using the Microsoft Kinect from the 3DRS dataset [15]: three objects with clustering and defined bounding boxes.

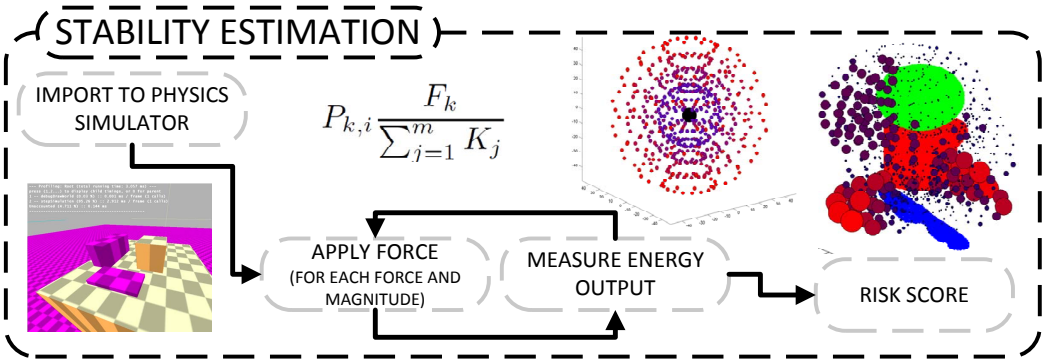


FIGURE 3.5: Stability estimation flow. Scene objects are imported into the physics simulation. Forces are applied from a sample of directions to each object in the scene, subject to (3.4). The energy output from each applied force is recorded. Simulations are repeated with forces of increased magnitude. For each object the resultant energy from each simulation is used to build a stability plot. The sum of all resultant energy defines the stability of the object and by extension its risk score.

With this voxel representation of the scene, clustering of the voxel volume can be applied. A number of different clustering algorithms were tested, using modified versions of the work presented in [50, 140]. A bounding box for each object cluster is defined, the dimensions of which are based on the returned clusters.

To represent the objects within a physics simulation, a range of bounding shape primitives can be used (e.g. box, cylinder, sphere). The shape primitive that, when fully encasing the cluster has the least empty voxels, is the one that best defines the object cluster. Additionally these bounding shapes must not intersect; as such a recursive process is applied reducing bounding boxes until no overlap is detected. The result is a preprocessed scene in which each detected object cluster is assigned its own bounding shape (Figure 3.4).

3.3.3 Stability

The proposed methodology for scene stability estimation is based on the use of Newtonian physics mechanics applied to the preprocessed scenes. To evaluate the stability of an object we replicate the application of forces from a variety of directions. Consequently, statistical analysis on the subjects of a simulation can be performed allowing the computation of the energy output from each applied force. An overview of this is presented in Figure 3.5.

Using 'collision shapes', in this case bounding boxes, the objects are recreated using simplistic primitives, which represent the overall shape. This reduces the computational

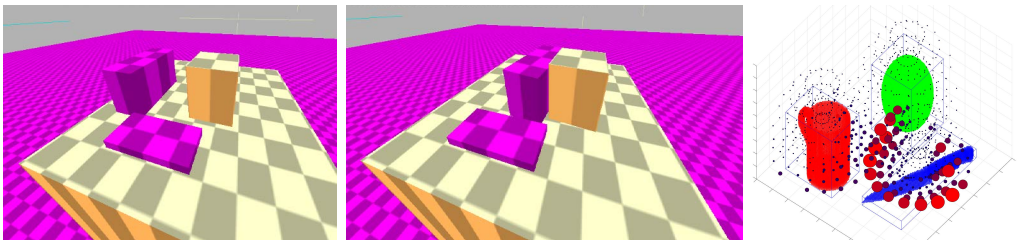


FIGURE 3.6: Stability evaluation process using Newtonian physics. (Left) Initial layout in the physics simulation. (Middle) Collision occurring during the simulation, and (Right) stability plot with the circles around the objects indicating the direction of instability with radius corresponding to the severity.

costs needed to emulate its behaviour whilst maintaining a reasonable level of accuracy. To simulate an object's behaviour; parameters such as position, size, mass, friction and angular dampening coefficients are attached to these shapes. The bounding shape calculated during preprocessing serves as the guidelines for the collision shape and also defines its position and size within the simulation (Figure 3.6).

The surface the objects are placed on within the simulation is defined using dimensions obtained during the plane removal process in preprocessing. Mass is defined by calculating the number of voxels within an object cluster and using the assumption that the object's material is given. However through the use of material estimation (such as BRDF function estimation [141, 142] or techniques such as visual vibrometry [143] as well as others [52, 144]), more accurate values for mass could be acquired to increase the realism of the simulations. Additionally with a defined material, the friction coefficients can also be better estimated and applied to the simulation. These techniques would be applied during preprocessing (Figure 3.3). However, this falls into a separate area of research and is not the goal of this work, as such the values used here are assumed to be provided.

Instability s as a result of a force k on a given object i is defined as the ratio of the applied force F_k over the summed kinetic energy K_j for all objects m in the scene. This is scaled by the possibility $\mathbf{P}_{k,i}$ of the force being applied.

$$s_{k,i} = \mathbf{P}_{k,i} \left(\frac{F_k}{\sum_{j=1}^{m} K_j} \Delta x \right) \tag{3.3}$$

where $K_j = \sum_{t=1}^{T} \frac{1}{2} M V_t^2$ represents the accumulated kinetic energy produced by the object j over time T as a result of the force k being applied during the simulation,

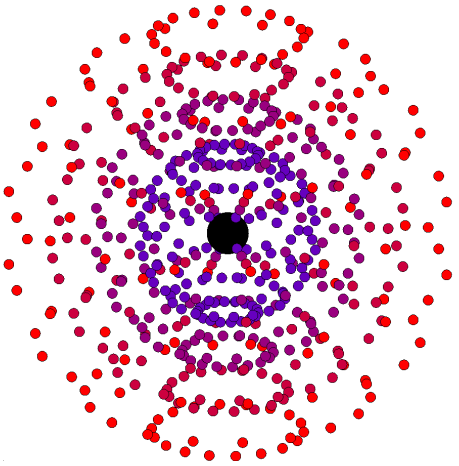


FIGURE 3.7: A visual representation of the force applied to an object. The black sphere represents the object and the small blue to red coloured spheres represent the direction the force is applied from. The distance away from the black sphere represented the magnitude of the force applied.

obtained using numerical integration. Here M represents mass and V the velocity of the object j at a given time t. Δx is an object's displacement but since the kinetic energy is calculated numerically over fixed length intervals, this value is equal to one. $\mathbf{P}_{k,i}$ represents the likelihood of a given force F_k being applied to object i. This is defined as whether the force could collide with the object without hitting first another entity within the scene. For example forces from below an object on a plane would collide with the surface first, therefore would not be considered.

$$\mathbf{P}_{k,i} = \begin{cases} 1, & if\ F_k\ directly\ collides\ with\ object\ i \\ 0, & otherwise \end{cases} \tag{3.4}$$

Forces of different strengths are applied to the center of each collision shape (object) during the simulation. The strength of these forces is widely sampled to ensure that objects of both large and small mass are affected and provide a measurable energy output. The direction of force is selected from a uniform sample over a sphere (Figure 3.7).

The resultant overall kinetic energy K for each object j is calculated. By analysing the amount of kinetic energy produced by each object for each force F, we can ascertain if, during the course of that simulation, an object has been dislodged from the surface or if other objects within a scene have been affected due to collision. By varying the strength

of force we build up a picture of how unstable an object is in its environment. The total instability S of a scene is given as the sum of the estimated instability s for each force k applied to each object j.

$$S = \sum_{k=1}^{r} \sum_{j=1}^{m} s_{k,j} \tag{3.5}$$

The outcome of this in regards to a risk score is such that an object (e.g. glass bottle) being placed at the center of a table will have a lower score than one placed at the edge (Figure 3.1).

3.3.4 Simulation Prediction as a Regression Problem

In Section 3.3.3, physics simulation is utilised to calculate the estimated energy output of an object from an applied force. This provides a realistic form of evaluation for the instability of objects within a scene. However this level of accuracy comes at a computational cost. Indeed the utilisation of real time simulation techniques in many fields puts a stressor on available hardware. As an example, the games industry which arguably drives the industry forward in terms of computational requirements and expected outputs, requires gaming environments to accurately simulate the physical principles of our world to create an immersive experience.

To help remedy this and provide a system that is able to run on hardware that may have limited processing power, the concept of simulation prediction is introduced. This is important within risk estimation, as with the expected domestic applications, processing power may be at a premium. Physical concepts such as acceleration, velocity or position over time, could be predicted using a few initial measurements and a set of training data. For example, instead of performing the complex physics simulations for an event's entire duration, only a few initial calculations over a very short amount of time could be applied and the remainder of the simulation predicted using the proposed prediction mechanism. This presents a huge computational saving at the cost of a minor decrease in accuracy.

Figure 3.8 outlines the prediction framework that incorporates the proposed methodology. The framework is divided up into two main areas; 'Background Process' where the

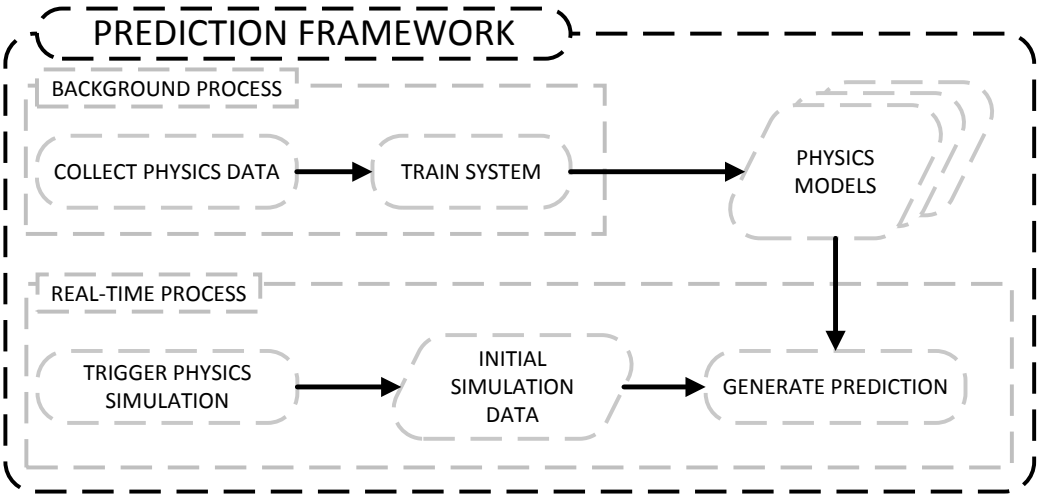


FIGURE 3.8: The proposed prediction framework.

training models are constantly updated and improved, and 'Real-Time Process' where the previously trained models are used in the prediction of that event. The framework is designed to be open ended with the ability to drop in different techniques during training, depending on the application. Importantly any suitable machine learning technique would be applicable, allowing for flexibility for the purpose.

The trained physics models will dictate the types of events that can be simulated. Simple events, where objects in a scene will not interact with others, are easily handled with the proposed framework. More complex events, where simple collisions with other objects are required are also modelled effectively. This is important in the analysis of object stability as the movement of an object in the scene may impact others, thus any prediction framework must be able to emulate this behaviour. A fundamental advantage here is the reuse of trained models; preventing the need to fully simulate similar events each time. This is very suitable for stability estimation as the application of forces is predictable.

To define a model of an object's movement in 3D, first the simulated track in terms of x, y and z coordinates per frame of the simulation is required. To enable comparison later, all simulated object trajectories must be aligned in the 3D space such that they all follow a uniform direction from a uniform point. Firstly the object's track is centred to the origin by applying a translation, this is based on the vector from the origin to the first point of the track. Secondly a rotation is applied to the track around the up axis (e.g. that which is perpendicular to the ground plane of the simulation) to ensure that all the simulated tracks are following a uniform direction. Importantly the translation

and rotation is applied to the track as a whole to maintain the deviations and local trends of each track.

Next the problem of physics simulation prediction is reformulated as a regression problem, utilising the advantages of Principal Component Regression (PCR). Regression tries to estimate the relationship between a dependent variable (location or acceleration) and independent variables (a selected initial energy or force under a selected direction). This analysis allows the prediction of the physics dynamics in the near future based on some initial observations. PCR has the advantage of overcoming the collinearity problem which occurs when two or more of the explanatory variables are highly correlated. This is dealt with by excluding low-variance principal components in the regression step. This has the secondary advantage of dimension reduction, as only a subset of the data's principle components are utilised in the regression model, due to the high correlation the prediction abilities of the model remain largely unaffected.

Firstly let the simulation data be represented by their dependent variables $\mathbf{Y}_{n\times 1} = (y_1, ..., y_n)^T$ and their independent variables $\mathbf{X}_{n\times p} = (x_1, ..., x_n)^T$ which, in this case, might be a vector indicating the direction and magnitude of an applied force or equivalently an obtained acceleration and position. The goal of PCR is to estimate a model such that for a given X_i, a value for Y_i can be estimated using a model $\mathbf{B}$ based on the all the sample data of X, with ϵ representing the error.

$$Y = X\mathbf{B} + \epsilon \tag{3.6}$$

$$Y = \begin{bmatrix} y_1 \\ \dots \\ y_n \end{bmatrix} \qquad X = \begin{bmatrix} x_{1,1} \dots x_{1,p} \\ \dots\dots\dots \\ x_{n,1} \dots x_{n,p} \end{bmatrix} \tag{3.7}$$

Initially Principal Component Analysis (PCA) is applied to X, resulting in a spectral decomposition of $X^T X$ in the form of $\mathbf{V}\Lambda\mathbf{V}^T$, where Λ is a $p\times p$ matrix of the eigenvalues and the columns of $\mathbf{V}$ relate to the eigenvectors. $\mathbf{V}$ contains an eigenvector for each p dimension, ordered such that the first eigenvector corresponds to the largest variance of the data contained in X. As such $\mathbf{V}_k$ represents the k principal axes onto which

the variance retained under projection is maximal for the data in X. In this case it is expected to be the x, y component of the track in 3D space. This approach generates a mapping function $\mathbf{V}_k$, from the high dimensional space to the low and vice versa.

$$W_k = X\mathbf{V}_k \tag{3.8}$$

Here W_k is the lower dimensionality representation of X subject to $\mathbf{V}_k$. Using W_k the ordinary least squares regression technique can be applied based on the response vector of Y.

$$\gamma_k = (W_k^T W_k)^{-1} W_k^T Y \tag{3.9}$$

Finally the PCR estimator for k principle components of X is given as

$$\mathbf{B} = \mathbf{V}_k \gamma_k \tag{3.10}$$

To create a model, training data in the form of simulated tracks is run through the preprocessing stages outlined in Section 3.3.2. PCR is applied for all objects and applied forces within a scene, producing a model for each trajectory and generating mappings from high to low dimensional space. The mappings serve the purpose of both converting a new test sample into a low dimensional representation and also for translating the model data back into high dimensional space.

For the prediction, a subset of data is simulated for each object in a test scene and goes through the same preprocessing stage. Test data is mapped to low dimensional space using each mapping available from the models. Using the resultant low dimensional representations of the test data, a matching process is run to find which model's low dimensional representation of it's simulation has the closest Euclidean distance from the test example. Once an appropriate model has been selected, the low dimensional representation of the test case is predicted using the PCR model and translated back

FIGURE 3.9: Some scenes of the new 3D Risk Scenes (3DRS) dataset [15] with the three levels of stability for each one.

to high dimensional space. This track represents the predicted movement in 3D of an object after a force has been applied. Using this prediction, the energy output for that force k for object i is calculated and added to that scenes instability value as outlined in 3.5.

3.4 Results

The following section outlines the experiment environment and subsequent evaluation of the methodologies outlined above. Firstly the stability estimation concept is reviewed, followed by the prediction method.

3.4.1 Experiment Environment

To effectively test the proposed stability estimation methodology the 3D Risk Scenes (3DRS) dataset is used. Using Microsoft's Kinect and Kinect Fusion [4], 42 indoor scenes (Figure 3.9 - 3.10) were captured. Scenes are constructed using a subset of the 3DRS datset's 27 household objects (Figure 3.11). All scenes were configured on a square table consisting of three objects per scene. In more detail, these 48 scenes are split into 16 different scenarios. Each scenario has three iterations that represent a different stability level based on the object's predefined locations, i.e. the objects are moved closer together on the plane within the scene (Figure 3.12).

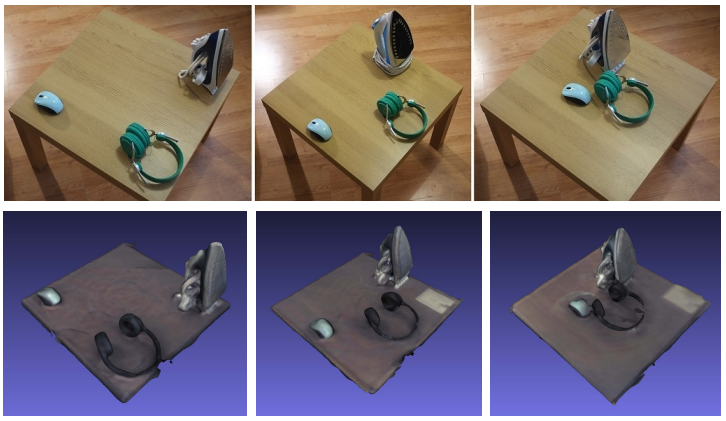


FIGURE 3.10: A scene from the new 3DRS dataset reconstructed using Kinect Fusion for the three levels of stability.

FIGURE 3.11: Some objects of the 3D Risk Scenes (3DRS) dataset.

In order to obtain the ground truth for each scene and to ensure that the parameters of the tests are fully controllable, the objects were manually placed on a surface at predefined locations. Each location as we can see in Figure 3.12, is represented by a different colour which corresponds to a specific stability-risk level.

Each scene is run though the preprocessing steps laid out in Section 3.3.2. For all cases a voxel volume representation is returned with a resolution of 256x256x256 voxels, representing an approximate volume of $50cm^3$. Any lower resolution and shape information about the object would be lost during voxelisation. The returned 3D reconstruction of a scene from Kinect Fusion has some preliminary smoothing and hole filling techniques applied and therefore any higher resolution would not significantly affect the overall performance. The resolution also has a direct impact on computation time for each stage and as such this represents a reasonable trade off for processing time against object detail.

Scene segmentation is part of the preprocessing stage and as such a number of tests were carried out to ascertain the most effective segmentation algorithm to use with the dataset. The segmentation algorithms evaluated included; K-means using a random

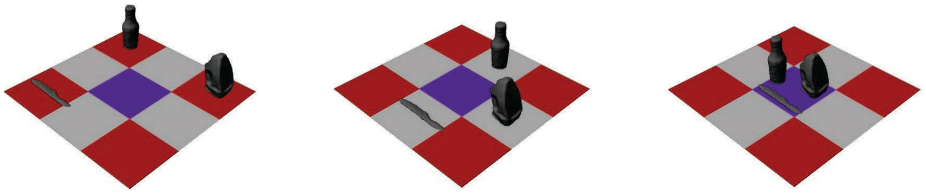


FIGURE 3.12: An example scenario with each of its iterations. The level of complexity and stability is increased. Left (Lvl 1), a simple layout with lower complexity but higher instability. Mid (Lvl 2), average complexity and instability. Right (Lvl 3), a complex layout with lower instability.

TABLE 3.1: Segmentation accuracy for all the levels of stability (Figure 3.12). Accuracy defined as the percentage of voxels assigned to the correct object cluster.

Stability	K-Means [50]	Mean Shift [140]	Distance
Lv1	98.86%	97.58%	86.45%
Lv2	86.26%	86.88%	83.32%
Lv3	82.87%	81.62%	78.17%
Overall	**89.33%**	88.69%	82.65%

preliminary clustering phase, Mean Shift with a bandwidth parameter found experimentally, and distance based clustering utilising predefined centroids. Ground truth was established manually and accuracy is defined as the percentage of voxels correctly assigned to their respective object cluster. The results of which are presented in Table 3.1. As the objects in the experiment environment do not touch, the object clusters are defined well enough that a predefined number of clusters is not required to achieve good segmentation. In the instances where voxels are assigned to the wrong object cluster, bounding shapes are still obtained based on the wrongful classification. However, due to the recursive reduction phase, the bounding shapes are iteratively reduced to a point where there is no longer any interaction between them.

The algorithms are evaluated on all scenarios and results are grouped according to stability level, which represents an increasing level of difficulty for the segmentation algorithms (Figure 3.12). Level one represents the objects placed at the maximum distance apart, with level three representing all three objects in close proximity. The k-means algorithm was found to be the most efficient at separating the objects across all the complexity-instability levels.

3.4.2 Stability Evaluation

To demonstrate the effectiveness of the proposed stability concept a proof of concept experiment was done. Three experiments were conducted in which an example bounding shape (cube) was passed to the physics simulation in three different positions on a surface

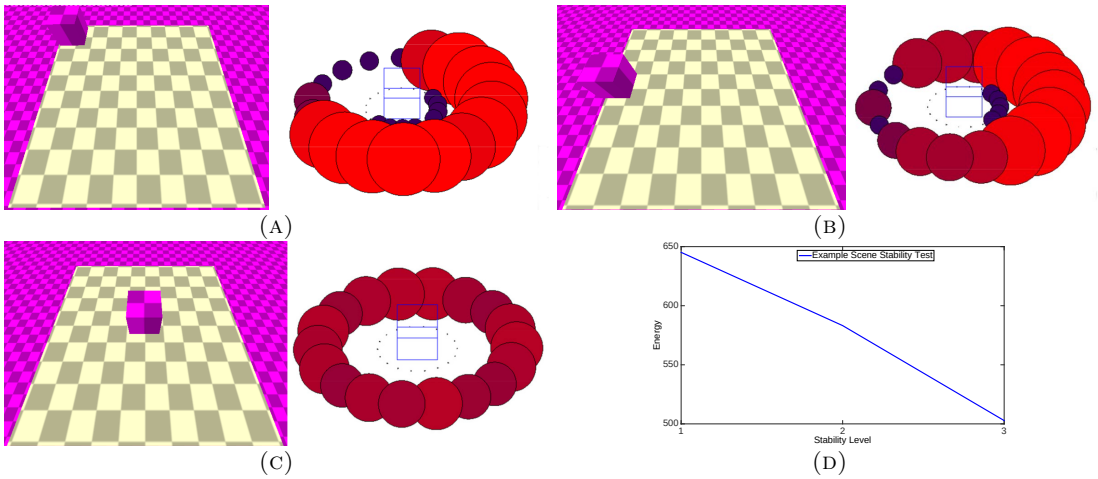


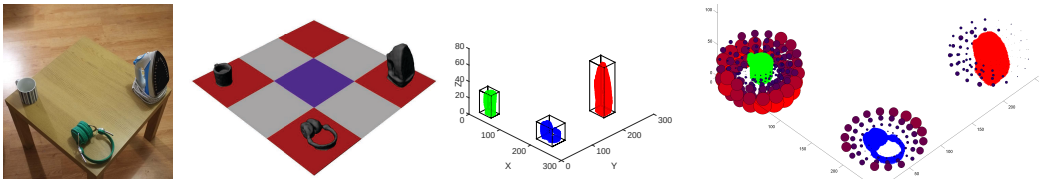
FIGURE 3.13: Example scene stability test. (A) Far left corner, (B) left side and (C) centered. (D) Scene energy per stability level in graph form. The larger the sphere the more energy output as a result of the force. Additionally emphasized by colouring, where red is a high energy output and blue a low.

plane (Figure 3.13 A, B & C). The simulation was run and the position and rotation values for the cube were recorded over time. The simulation software employed utilises the Bullet 3D Real-Time Multi-physics Library [145]. To visualise the data we position spheres to represent the source and magnitude of the applied force; the further away from an object a sphere is, the larger the magnitude of force it represents. The colour and size of each sphere represent the resultant instability. The larger and more red a sphere the higher the energy output as a result of the force applied from that direction. In these examples, forces were applied from 18 points around the object, each with two levels of magnitude. Forces applied from a direction that would push the object off the table result in the largest energy output, thus represent higher instability.

As with the 3DRS dataset, this example scene has three levels of stability. As the object comes towards the centre of the scene we can see that the energy output decreases (Figure 3.13 D). This follows the logical assumption that objects at the centre of a table are less risky than those at the edge or corner.

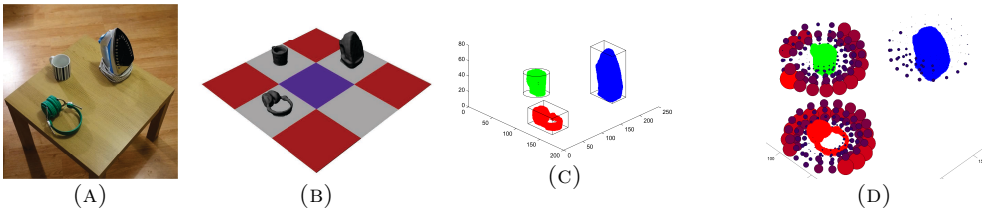
To further evaluate this, the stability of 48 real scenes from the 3DRS dataset (16 scenarios each with three stability levels) were also analysed. For these experiments, force was applied from points (directions), uniformly sampled along a sphere, with four levels of magnitude (Figure 3.7). As each scene contains more that one object, and all objects in a scene are represented in a simulation at the same time, the effect of collisions between the objects is also taken into account. This is visible on the stability plots, especially those of the small objects such as the knife or mouse. For the simulation

an object's friction coefficient was set to 1, while the angular dampening coefficient was experimentally selected to be 0.4. These values are available for the objects within the dataset however, as material estimation is not included within the framework, global values were utilised at this stage. As all objects in the dataset were assigned the same values there is little difference to the results when changed. As such the values chosen have been done so to produce realistic movement for all objects from the dataset in the simulation and according to the suggested values of the physics engine. To maintain an autonomous system a rudimentary measure of mass is given by the number of voxels that each object cluster contains. The scenes' overall stability was quantified according to (3.3), (3.4), and (3.5).



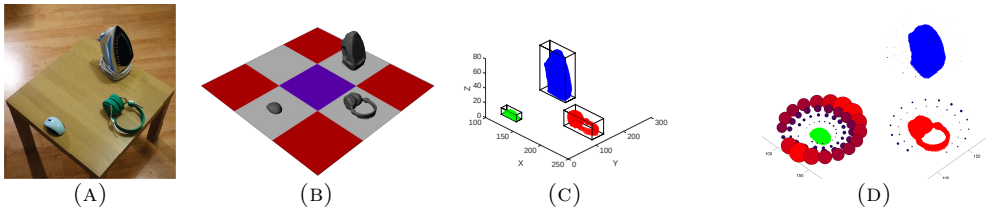
(A) (B) (C) (D)

FIGURE 3.14: (A) Real image, (B) digitized (C) voxelised and clustered. (D) Force plot.



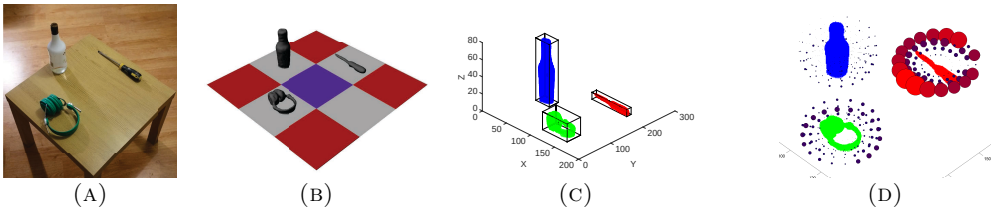
(A) (B) (C) (D)

FIGURE 3.15: (A) Real image, (B) digitized (C) voxelised and clustered. (D) Force plot.



(A) (B) (C) (D)

FIGURE 3.16: (A) Real image, (B) digitized (C) voxelised and clustered. (D) Force plot.



(A) (B) (C) (D)

FIGURE 3.17: (A) Real image, (B) digitized (C) voxelized and clustered. (D) Force plot.

In Figure 3.14 - 3.17 example instability results are shown. Regarding the collision shapes, in these cases a cube shape was used to approximate the objects of each scene. Figure 3.14 - 3.17 (A) shows some of the real test scenes, (B) shows the 3D mesh models

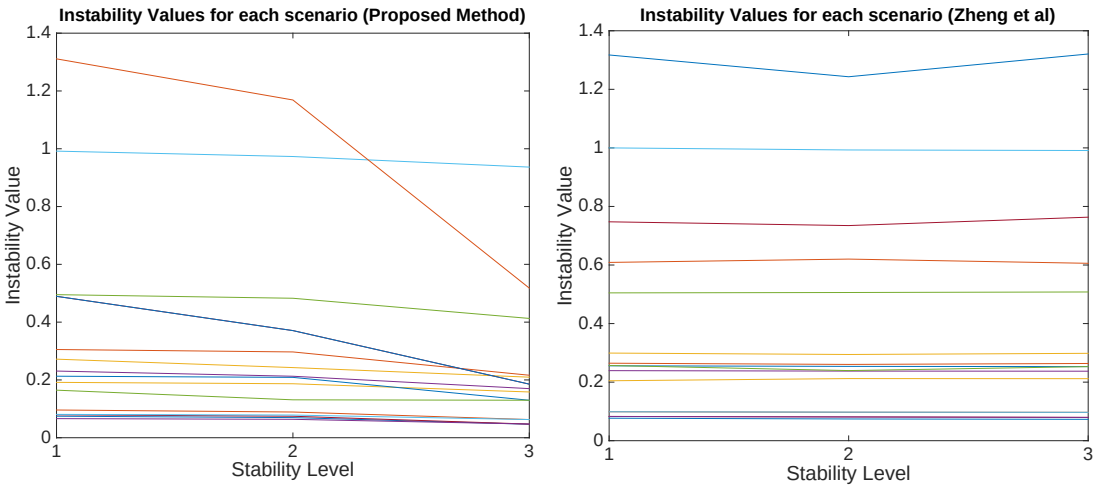


FIGURE 3.18: Instability graph. (A) Proposed method, (B) work presented in [16]. Lines correspond to the 16 scenarios. Instability value obtained using (3.5), measured across the three different stability levels. Higher the instability value the less stable the scene is.

TABLE 3.2: Average instability values over all the scenarios at each stability level for the proposed method and the work presented in [16].

Method	Lv1	Lv2	Lv3
Proposed (Mean)	0.3469	0.3138	0.2199
Proposed (STD)	(0.3524)	(0.3227)	(0.2314)
Zheng [16] (Mean)	0.3837	0.3766	0.3833
Zheng [16] (STD)	(0.3646)	(0.3519)	(0.3659)

of the objects after capture using Kinect Fusion, (C) shows the scene segmentation results and the obtained bounding boxes and in (D), the stability plots with spheres around the objects indicating, with their location, the possible direction of instability and, with their radius/size, the instability level.

A comparison was made of the proposed stability estimation approach with the current state of the art [16]. Both methods were tested on the same scenes and the results indicate that the proposed method, which takes into account the possibility that objects may collide with each other, produced more realistic estimates. In Table 3.2 the obtained average stability values for the evaluated 48 scenes are given, both for the proposed method and the work presented in [16]. Each scenario becomes more compact and centralised as the stability level changes. Observing the results we can see the effect that grouping the objects together and moving them closer to the centre of the surface has. With the proposed method, the risk score is reduced (Figure 3.18 A) however this effect is not reflected in the technique proposed in [16]. This observed reduction in risk follows the logical assumption that those items in the centre of a table are more stable that those at the edge.

It can also be observed, from the stability plots, that additional stability is gained as objects are placed in close proximity to one another since their potential collisions will reduce the overall instability. However the increase in stability is not always uniform, this is, in part, down to the differing size of objects in each scene. The properties of the objects, such as size, mass, and shape will all have an impact on how the stability of a scene changes. For example, a scene with one larger object and two smaller, will have a distinctly different stability plot to one where the objects are of a more uniform size and mass. This is due to the stabilizing effect the larger object would have on the smaller.

Using this proposed method a stability value S can be defined for use within the Risk Estimation framework. The stability measure can be given at an object level or for the scene as a whole. Using the obtained measurements on a per object basis, a representation of the stability for a scene from the 3DRS dataset is given (Figure 3.19).

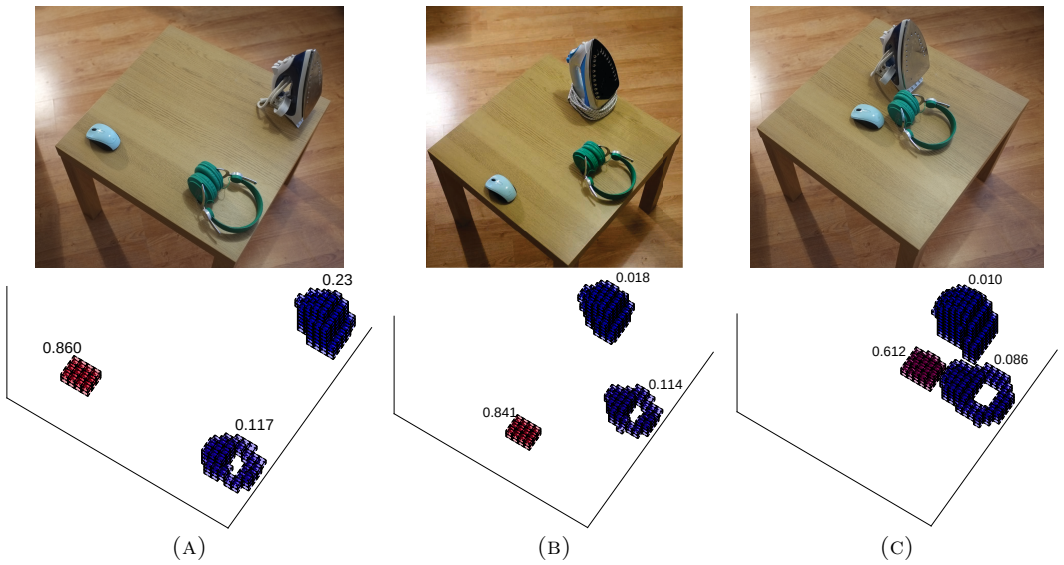


FIGURE 3.19: Illustration of instability per iteration of an example scene. As the objects get closer together and further from the edges of the table the instability and subsequent risk scroe goes down

Table 3.3 gives the risk score for instability for each scene in the 3DRS dataset according to 3.2 and 3.5. Results are given for each scenario and each of its three levels. The higher the level, the closer together and more central the objects within the scene are located. As such, the expected instability value should decrease as the level increases.

3.4.3 Predictive Physics Evaluation

In order to evaluate the performance of the prediction framework, an evaluation is conducted using the stability estimation simulation data created using the Bullet open

TABLE 3.3: Risk score for instability taken from physics simulation data using the 3DRS dataset, given for each scenario and each level.

Scene	Lv1	Lv2	Lv3
1	0.10	0.09	0.06
2	0.40	0.39	0.28
3	0.25	0.24	0.21
4	0.30	0.28	0.22
5	0.64	0.63	0.54
6	1.29	1.26	1.22
7	0.10	0.10	0.06
8	0.28	0.27	0.17
9	0.12	0.12	0.08
10	0.35	0.32	0.27
11	0.09	0.08	0.06
12	0.21	0.17	0.17
13	0.10	0.10	0.08
14	0.64	0.48	0.24
15	0.64	0.48	0.24
16	1.70	1.52	0.67
Avg.	0.45	0.41	0.29

source physics engine. Using the dimensionality reduction methodology, the suitability of this technique as part of the Risk Estimation framework is illustrated. The stability estimation data contains the position and rotation data for each of the objects, per applied force, for a recorded 600 frames. In many cases, only the object which the force was applied to moves within the scene. As such for the testing of the prediction framework, only objects which have an energy output greater than zero were included in the evaluation data. The focus of the evaluation is to generate realistic object tracks in the 3D environment such that stability estimation, based on predicted data, remains similar to that of fully simulated.

After preparing the stability estimation data, as outlined in Section 3.3.4, testing was carried out using the 'Leave One Out' paradigm. As such the data was split; 10% of the simulation data was used to test, with the remaining 90% utilised as training data to create an 'objects' model from which predictions could be based. The tests were repeated ten times to cover all data. Prediction and subsequent matching was based on the first 10% (60 frames) of each object's track. The dimensionality reduction process reduces the input data to two dimensions.

For the created 'objects' model, results are given firstly in terms of the accuracy of the model itself, i.e how similar is the modelled data to the original simulated track data. Secondly the test accuracy is given, in which the predicted track for a test sample is

measured against that test's ground truth. The accuracy of a model was defined using the Euclidean distance between the modelled track of each object and the original source track from the simulation data. These values are scaled according to the distance which that object travels and is given as an average per frame. Table 3.4 demonstrates the model's accuracy in replicating the defined tracks of the objects using dimensionality reduction modeling technique. The results are broken up into average error per force magnitude. As the error measure is scaled according to the distance an object travels, the magnitude of the force should have relatively little impact on the final error measures.

TABLE 3.4: Model accuracy of the dimensionality reduction methodology on the stability estimation data.

Force Magnitude Lv	Lv1	Lv2	Lv3	Lv4
PCR Model Error	0.1083	0.0801	0.0484	0.0324

Figure 3.20 demonstrates how the error in the model propagates over time. A predictable compounded error over the course of the prediction is seen. The sharp increase in error within the first $20 - 30\%$, can be attributed to the fact that objects within a scene will often not stay in motion for the full 600 frames. Additionally the increase in error over the course of the model is due to a propagation effect which occurs when the trajectory of the prediction differers slightly from the original simulation. As such, as a simulation continues, the simulated and predicted tracks deviate further, increasing the error.

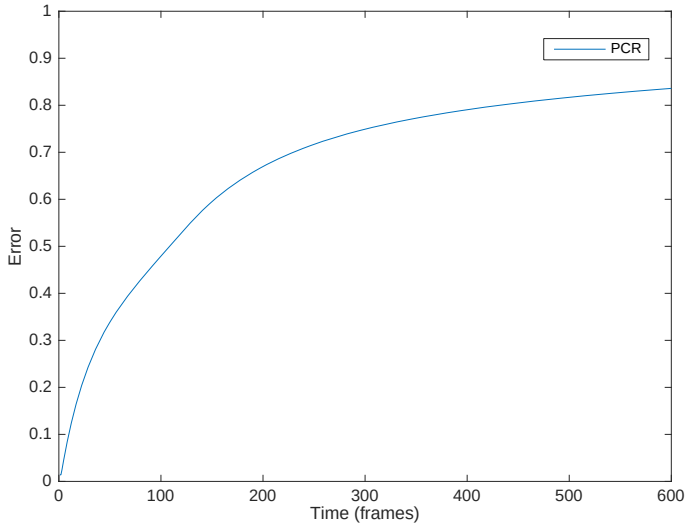


FIGURE 3.20: Model error per frame across all tests for the dimensionality reduction technique.

Table 3.5 demonstrates the accuracy of each model in predicting the defined tracks of the objects. Error levels here are very close to the model error rates. This implies that there is little additional error accumulated through the prediction process and rather it is the modeling technique that produces the error. This is in part due to the simplistic

nature of the scene, as the applied forces are coming from the same uniform directions, object interaction are highly predictable.

TABLE 3.5: Test accuracy of the dimensionality reduction methodology on the stability estimation data.

Force Magnitude Lv	Lv1	Lv2	Lv3	Lv4
PCR Test Error	0.1099	0.0817	0.0496	0.0331

Figure 3.21, is the average error of predicted tracks on unseen test samples against the ground truth. It can be again seen here that there is very little difference between the models prediction accuracy against its modelling accuracy. This demonstrates that there is almost no additional error generated by the prediction method and that the modelling process itself is the cause of this error.

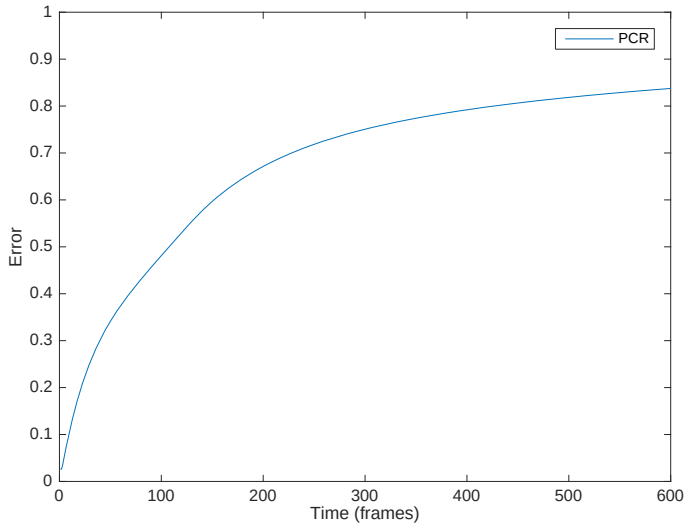
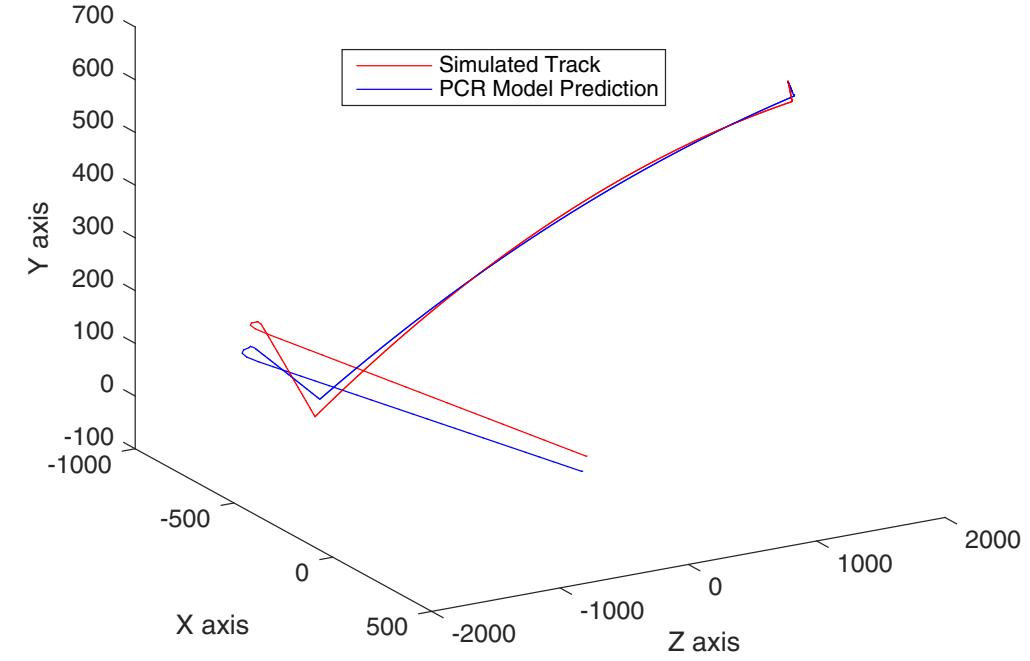


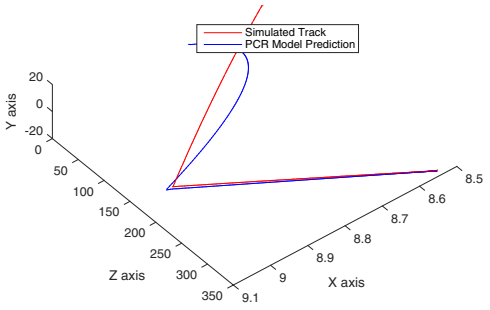
FIGURE 3.21: Prediction error per frame across all tests for the dimensionality reduction technique.

Figures 3.22 demonstrate some sample scenes in which a force is applied to an object in the scene, with both the simulated object track and models predicted track shown within the same figure. In the examples it can be seen how the dimensionality reduction method handles situations where sharp changes in direction are found.

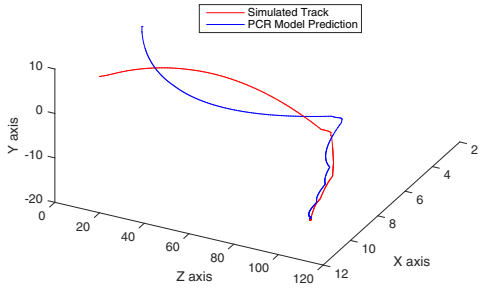
As shown, the model has the capability to generate accurate predictions using only the first ten percent of simulation data. With this in mind the energy output for all the scenes in the 3DRS dataset was calculated using the model data and is shown in Table 3.6. As can be seen from the results, in only four cases the expected decreased instability over the three levels is shown. This highlights that though the method is capable of creating accurate predictions, the matching process used to select a prediction model is not consistent. This could be due to the use of only the initial ten percent of simulated



(A)



(B)



(C)

FIGURE 3.22: Visualisations of an object's simulated track (red) and its modelled track (blue) using the dimensionality reduction technique. Examples taken from the stability estimation data.

data. Figure 3.23 further highlights this by comparing the average instability scores for all 16 scenes across the three stability levels for the fully simulated and predicted methodologies.

As a demonstration of the computational advantages of this solution, an analysis was done at runtime of the CPU cycle speeds (ms) for physics simulation using the PhysX engine. Tests were carried out using the Unity Development environment with the timings generated and extracted from the built in profiler tools. Unity uses the term

TABLE 3.6: Risk score for instability taken from model predictions using the 3DRS dataset, given for each scenario and each level.

Scene	Lv1	Lv2	Lv3
1	2.956	3.145	1.575
2	0.002	0.003	0.003
3	0.003	0.003	0.004
4	2.319	2.685	2.029
5	1.119	1.731	2.090
6	1.615	1.262	1.077
7	3.478	3.246	1.936
8	1.323	1.215	0.833
9	2.680	3.541	2.600
10	1.077	3.478	3.246
11	3.395	3.192	1.936
12	0.004	0.004	0.005
13	2.688	2.941	2.643
14	1.488	1.898	2.188
15	0.079	0.120	0.492
16	0.487	0.560	0.370
Avg.	1.545	1.814	1.439

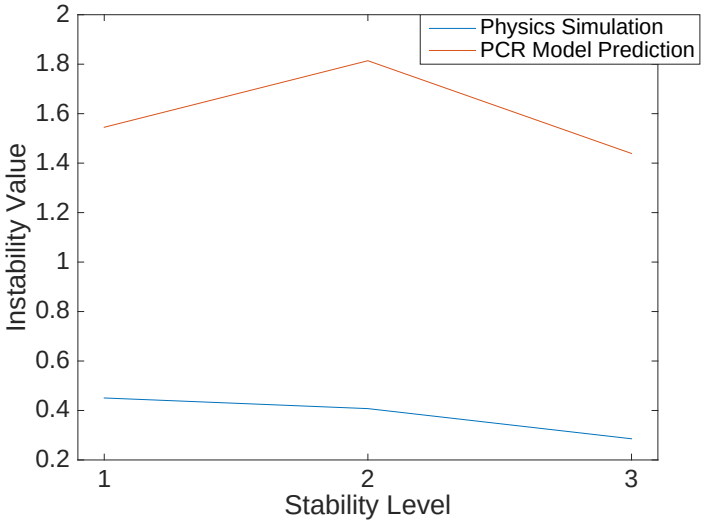


FIGURE 3.23: Average instability values for each stability level for the 3DRS risk scenes. Blue represents the instability values generated from full physics simulation. Red represents the instability values generated as a result of the prediction mechanism.

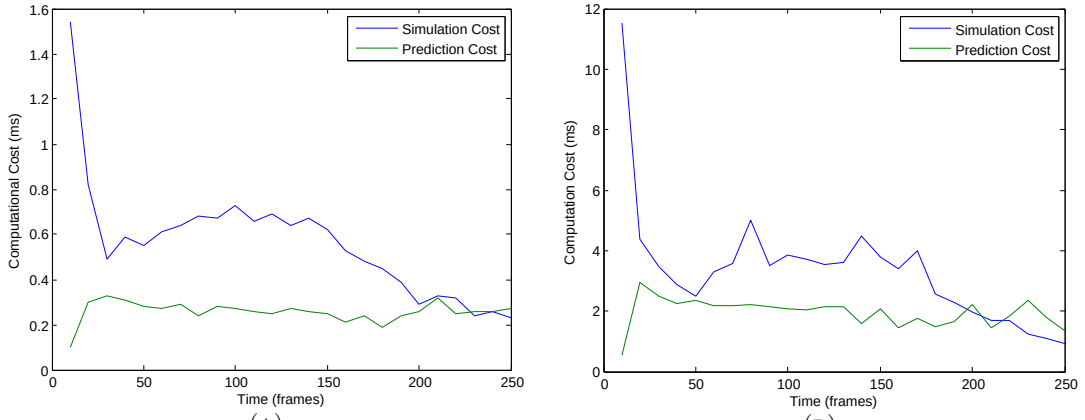
'kinematic' for object movement which is *not* calculated using the physics engine; as such during predictions objects are set to this state and positions and rotations are assigned each frame through scripting. Whilst simulation is in effect, all objects are set to 'non kinematic', where movement *is* handled by the physics engine. In this example a multi object explosion event has been created within the Unity environment and a model trained for the movement of these objects within the scene. In this case the event does not have a set duration, rather is considered finished once the amount of movement

in a scene drops below a threshold. As such this has been taken into account and computational cost has only been considered up to this point, in this case when there are no further interactions between objects and they have come to rest. Although in the a stability estimation implementation, visualisation of the outcomes of the simulation may not be required, this demonstrates both the computational saving and that in principle this method has applications in graphics and visualisation environments.

	100 objects	800 objects
Simulation	14.12ms	84.06ms
Prediction	6.46ms	48.70ms

TABLE 3.7: Analysis of the total computational cost for the physics engine whilst simulating a scene against running a prediction.

Tests were carried out on a single 100 cube scenario and a multiple instance 800 cube scene, with measurements being taken at 10 frame intervals. Figure 3.24 demonstrates the plot of those measurements and outlines that with our methodology a considerable reduction in computational time is achieved. Table 3.7 further outlines these savings, suggesting that the savings are as much as $42 - 54\%$ depending on scenario.



(A) (B)

FIGURE 3.24: Computational time required for physics simulation during: (A) a standard 100 object scenario and (B) an 800 object scenario. Blue represents the cost during simulation and red when prediction is used (scripting).

3.4.4 Conclusion

In this chapter the problem of evaluating risk is addressed through the introduction of the Risk Estimation framework. Allowing any element of measurable risk, the Risk Estimation framework presents a flexible and open ended approach to the concept of quantifying risk. In addition the problem of context is reviewed and a solution presented in the form of element weighting, allowing for more relevant measures of risk to be given

more precedence when producing a final risk score for an environment. The first element of measurable risk is presented in the form of stability estimation. Using simulation techniques to apply forces to a preprocessed 3D scene, a picture of instability can be built up, so that those objects which are placed in more unstable positions can be highlighted. Evaluation is conducted using the 3D Risk Scenes (3DRS) dataset which is specifically designed to provide scenes and objects that can be used for risk evaluation, and provides a benchmark in the scene analysis for risk area of research.

Additionally a method designed to reduce the computational cost of simulation is presented which uses the concept of prediction based on trained models. Using regression and dimensionality reduction techniques, simulated data is modelled and used as a basis to make predications for new unseen data. Evaluation is performed on the same stability estimation data. From the obtained results it can be observed that the prediction method presented has the potential for accurate predictions, additional refinement is needed in the model selection process to improve this accuracy. However given the advantages of the reduced overall computational cost, there is considerable benefit to be gained with its use within the Risk Estimation framework.

3.5 Discussion

With the introduction of the Risk Estimation framework, the foundation is laid for the evaluation of risk. Currently the proposed measure of stability provides a useful and generic idea of risk in an environment but more detail is required concerning the hazardousness of the objects themselves. The ability to assess the risk properties of the objects is crucial. For example having a large knife placed near the edge of a table poses a far greater hazard than a mug in the same position. As such, methods are required to identify the potential hazards that these objects may have. One approach to this would be the identification of the object itself, this could then be cross referenced with a database of risk objects to identify a risk score for use in the framework. However this approach opens itself up to other high level recognition problems such as classification granularity, i.e there are many types of knife, some more hazardous than others.

As such a lower level approach is needed where object characteristics themselves are recognised. Identification of sharp edges or other hazard features would avoid the need

to recognise the object itself and ensures the system is capable of identifying hazards of unseen objects.

Chapter 4

Object Risk Estimation and Hazard Elements

4.1 Introduction

The defined Risk Estimation framework allows the use of any measurable element of risk to quantify how hazardous a given scene is. So far the stability of the objects within a 3D scene have been analysed. However to better understand the hazards within a scene, classification of the objects themselves is required. Classification can be achieved in a number of ways; firstly by identification of what the object is, i.e this unknown object *is* a knife. Alternatively identification of the properties of the object, i.e this unknown object *has* an area of sharpness. This differentiation is key as it helps to simplify a complex problem. Rather than develop a method capable of identifying all known risky objects; a simpler model of risk properties can be created which allows unknown objects to be analysed based on a limited set of known priors.

The risk of an object can be split into many 'hazard' features. Within this work the term feature is used to relate to an actual physical property of an object (e.g. sharp, pointed) as opposed to the general computer vision definition, in which it describes an element of a scene. Hazard features represent an object's local properties; as an example with the use of a thermal camera, an object's temperature can be found and used as a hazard feature. The identification of an object's size and material can give an estimate of weight which again could constitute a potential hazard. Within this chapter the identification

FIGURE 4.1: Scenes of objects with intrinsic properties (e.g. sharp, pointed) and the goal identification of risky (red) objects versus safe (blue).

of shape related hazard features is addressed, using scene analysis and machine learning techniques and with a focus on classifying hazardous and safe household objects (Figure 4.1).

The use of 3D descriptors for identifying the object properties that relate to risk, is an emerging area of research. Many 3D shape descriptors can be found with applications in the computer vision and scene analysis areas of research and are highlighted in Section 2.2.1. Due to this absence of specifically designed features for this work's purpose, two new feature descriptors are introduced.

Dalal and Triggs created their Histograms of Orientated Gradients (HOG) feature for use in human detection in 2D images. Since that time HOG has been widely used in the 2D scene analysis area of research for many object recognition and detection applications. Using the principles of this technique and applying the process in 3D on voxelized data, the 3D Voxel HOG feature is introduced. The purpose of this feature is to create a consistent representation of object shape properties, such as blades or points with a classification of either hazardous or not. 3D VHOG works at a local level, analysing sections of an object rather than the object itself. Analysis is not only restricted to the surface of the object, 3D VHOG can also analyse internal structures when density data is provided. Combining this with a machine learning technique, a robust model can be created that can recognise these low level properties in new unknown objects.

In addition to this feature the Physics Behaviour Feature (PBF) is presented. This feature describes an object by the way it behaves when a force is applied, i.e the way the object falls through the air or interacts with the floor. Through the use of physics simulation data, the PBF can provide an informative descriptor for shape properties. The PBF works on an object level, providing a classification for the object itself. Utilising

both of these descriptors and their subsequent models, a risk score can be returned based on the confidence of classification.

One of the difficulties when performing classification tasks utilising feature descriptors, is the effect that outliers can have on the training process. The presence of outliers in the data can effect the robustness of a model by skewing what that model considers as a valid representation of a positive example. To remedy this process a robust kernel is introduced which transfers the desired 3D descriptor into a complex feature space and reduces the effect that those outliers have on the final training data. Using this process increases the accuracy of the 3D VHOG and PBF features and also shows an improvement on other common 3D feature descriptors.

Another key problem that asserts itself during classification tasks pertains to the model training stage. In this work Adaboost is used as the machine learning algorithm, one of the drawbacks to this method is the training time when using large feature vectors. To address this problem a complex variant of Adaboost is introduced which dramatically reduces the computational time and required number of training iterations. With the desired application of this work being for a domestic setting, the retraining and updating of classification models must be achievable on domestic scale computer hardware. Complex Adaboost helps to maintain this concept with similar levels of accuracy to traditional methods.

Within this chapter the following contributions are presented. Firstly the Risk Estimation framework is redefined to include the hazard features element. Hazard features are investigated and two new feature descriptors are introduced which look to classify hazardous properties of 3D objects. The first introduced feature descriptor is 3D Voxel HOG (3D VHOG) which extends the well known 2D Histogram or Orientated Gradients (HOG) into the third dimension. Secondly the Physics Behaviour Feature (PBF) which utilises physics simulation and the way in which objects react to applied forces to define the presence of hazardous features. Evaluation of the effectiveness of these methods is performed using the 3DRS dataset.

A method to improve the accuracy of the proposed features as well as others in the scene analysis field is also presented in the form of the Robust Kernel for 3D descriptors. The aim of this filter is to reduce the effect that outliers have in the training of machine learning models and improve the overall recognition rates during classification.

Finally an extension to the Adaboost [33] methodology, in the form of complex and hyper complex variants, is proposed which reduce the computational cost of training the classification models used in scene analysis.

4.2 Related Work

Due to the nature of the proposed work in this chapter there are no specifically designed feature descriptors that aim to model the concept of risk properties or their associated shapes. The most similar work that utilised the principles of HOG into their descriptor is that of Scherer et al [17] who does gradient computation in 3D using a convoluted distance field. This provides an effective way of calculating the magnitudes of the gradients, scoring them highly when localised near a surface of a model (local maxima). However their method also scores highly those at local minima creating additional artifacts within the data. As such this particular implementation is unsuitable for local feature recognition.

Due to the lack of specific comparable work, classic features are also evaluated for the task of hazard feature estimation. These include 3D HOG [68], 3D SIFT [146, 147], 3D Harris [148] and FAST 3D [149].

4.3 Methodology: Hazard elements

The following section outlines the Risk Estimation framework and in detail the additions of the hazard feature descriptors; robust kernel for 3D descriptors and the Complex and Hyper-Complex Adaboost methodologies.

4.3.1 Risk Estimation Framework

In Figure 4.2 an overview of the whole methodology is illustrated, outlining the end to end solution and where each of the proposed techniques fit. Initially the given scene is preprocessed to provide individual object clusters as per Section 3.3.2. Using these object clusters, the stability of each object is estimated providing one element of the risk score. The hazard features of each object cluster are then analysed, using the 3D Voxel

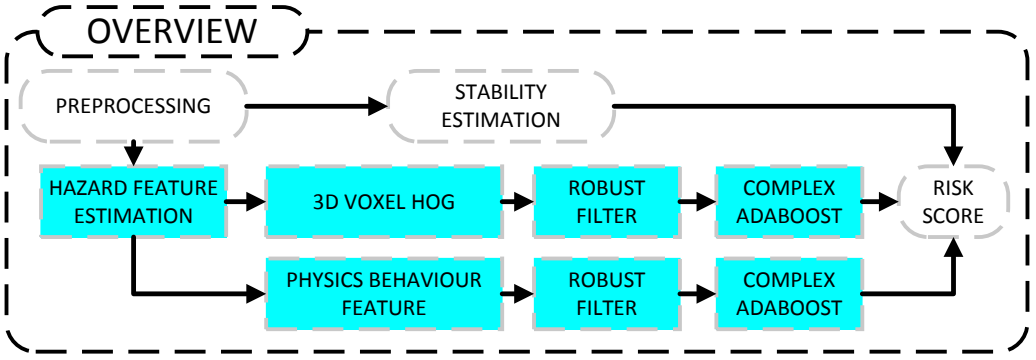


FIGURE 4.2: The overall methodology for the Risk Estimation framework, with each of the newly proposed methodologies highlighted.

HOG and the Physics Behaviour Feature, the results of which are used as the second element of the risk score.

With this additional proposed element, the Risk Estimation framework (3.1) is redefined, such that $n = 2$ where $w_1 e_1 = w_S S$ and $w_2 e_2 = w_H H$. As such the cumulative risk score R is now a function of the weighted elements of stability S and the hazard features H.

$$R = w_S S + w_H H \tag{4.1}$$

4.3.2 Physics Behaviour Feature (PBF)

Using the behaviour of an object within a simulation environment as a feature descriptor is a novel concept. Based on the data generated from a physics simulation a feature vector can be constructed and a classification made relevant to its risk. The essence of the methodology is to define a feature descriptor that describes how each individual object acts when a force is applied. In Figure 4.3, an overview of how this feature is incorporated into the Risk Estimation framework is presented.

Once preprocessing has been performed, as outlined in Section 3.3.2, an individual bounding shape for an object is passed to the physics engine. The goal is to take a single force from a fixed direction with a fixed magnitude and apply it to each individual object. The proposed feature descriptor is made up of the resultant simulation output data with reduced dimensionality.

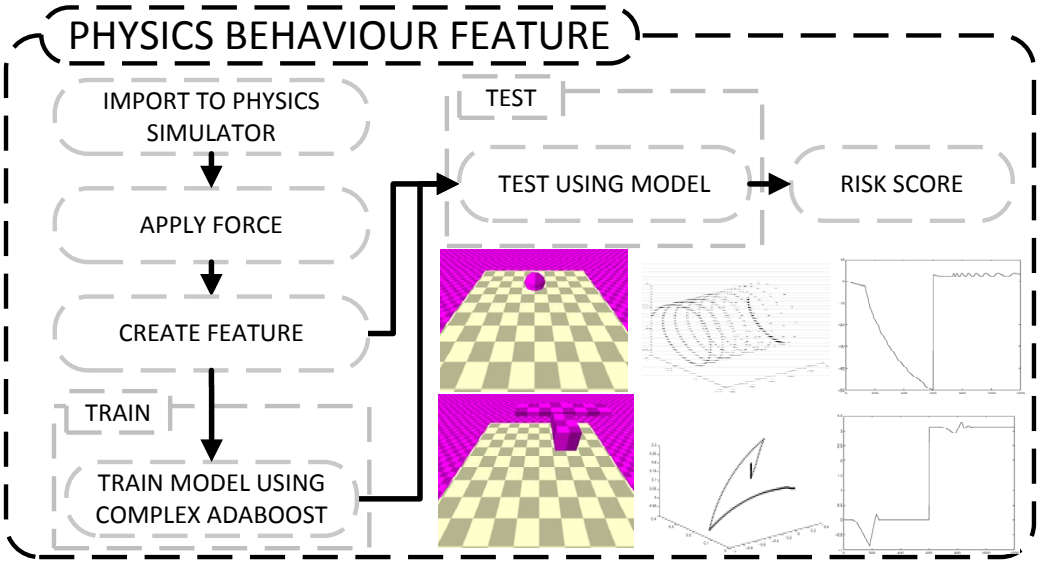


FIGURE 4.3: Physics Behaviour Feature (PBF) flow. Initially an object is imported into the simulation environment. A single force is applied to the object and the position and rotation information is recorded. A feature vector is constructed and a model trained using Adaboost. The process is repeated with a new unknown object and, using the previously defined model, a classification as either hazardous or safe is returned.

For a given object x, force is applied to its bounding shape and its angular velocity ω (in terms of x, y, z) recorded over the duration of the simulation time t. A feature vector is constructed from this data utilising dimensionality reduction to reduce three dimensions to two. The data is then sampled to reduce the length of the final vector. The resultant feature vector represents the physical shape characteristics and properties of an object in a scene.

$$\vec{x}^{\omega} = \{\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_t\} \tag{4.2}$$

Figure 4.4 (A - D) demonstrates this process. Firstly (A), illustrates the simulation process in which the bounding shape of the object has a force applied and the object's position and rotation recorded for each frame of the simulation. (B) is a plot of that object's movement through the scene in 3D, (C) is the two dimensional representation of that data and finally (D) being the sampled version of the data and the final feature vector for training.

These features are used to create a decision model from supervised learning. Each object is assigned a ground truth value of either hazardous or not from which training and evaluation is based. The subsequent model returns the same binary classification defining

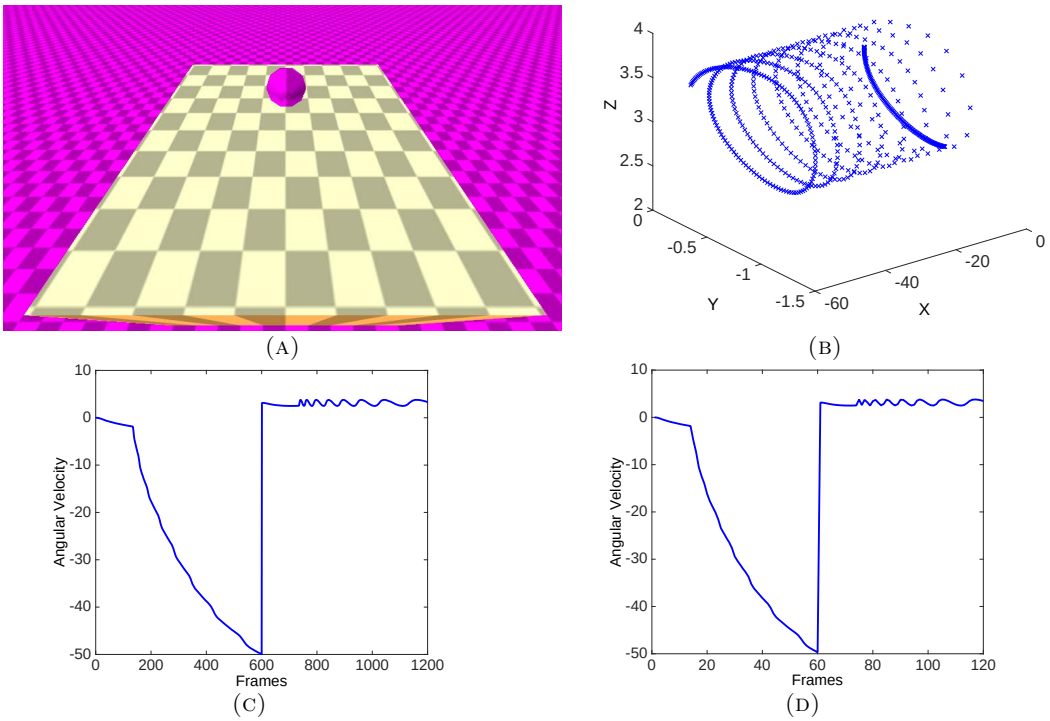


FIGURE 4.4: Physics Behaviour Feature, overview of the feature extraction process. (A) Simulation run on object bounding shape, angular velocity captured per frame, (B) the 3D plot of collected data, (C) data reduced into 2D space and (D) down-sampled to the final feature vector (ω) without any significant loss of information.

if the object is hazardous or not. A confidence score based on the model's assessment can be used as a weighting to the binary classification. These values contribute to the hazard features risk element as specified in (4.1).

4.3.3 3D Voxel HOG

To identify the properties of an object, identifiers are required that allow us to differentiate between hazardous objects from safe objects. In the Risk Estimation framework, rather than focusing on object identification, the recognition of hazard attributes is made the core problem. A novel classification problem of recognising sharp and pointed areas in a scene (hazard features) is introduced. The overall proposed classification approach for hazard areas in a scene is shown in (Figure 4.5). To achieve this a novel descriptor, 3D Voxel HOG, is introduced which extends the original Histogram of Oriented Gradients for use in the third dimension. 3D VHOG is suitable for recognition of local shape characteristics and additionally has the advantage of considering an objects' density.

Traditional Histogram of Orientated Gradients (HOG) [63] applies a gradient vector to each pixel in an image in either one or both of the horizontal and vertical directions.

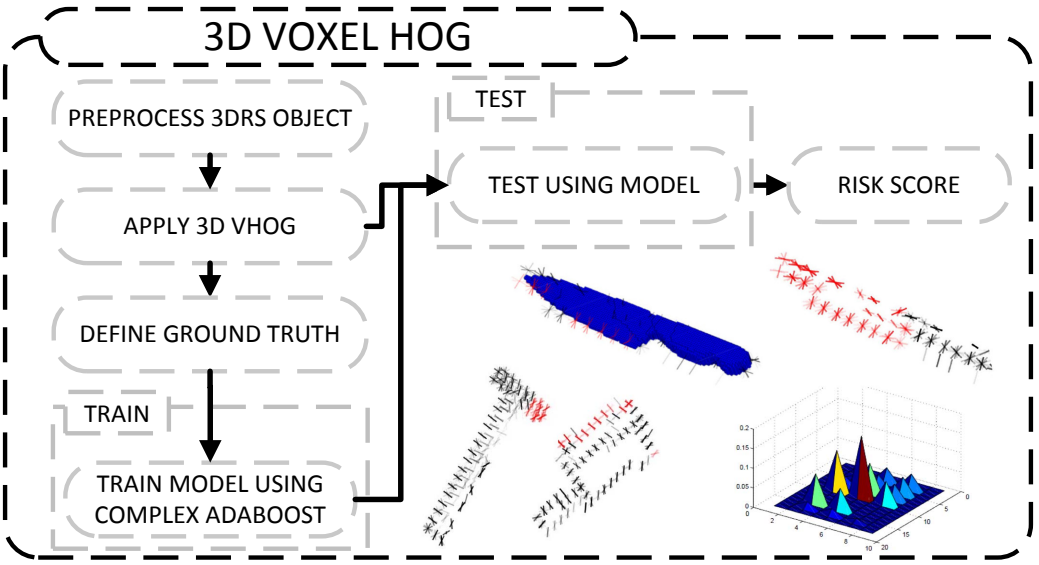


FIGURE 4.5: Overview of the proposed 3D Voxel HOG methodology. Each preprocessed object is represented using the 3D VHOG feature descriptor. A classification model is trained using Adaboost and used to test new unknown samples. Based on the classification, a risk score is derived for that object.

The image is then divided into overlaying blocks which in turn are made of a number of cells containing a set number of pixels. For each cell, a histogram is created with evenly spaced bins representing gradient angles. Each pixel's gradient angle votes for a bin, with the contributed value being weighted in some way (usually utilising the gradient magnitude). Finally each block of cell histograms is normalised locally to reduce the impact of changes in illumination and a concatenation of the histogram values is used as the final feature vector.

For the 3D Voxel HOG feature, this process is extended to the third dimension through the use of voxels. The process begins by breaking the preprocessed voxel volume up into set feature blocks f comprised of a number of cubic 3D cells c, which in turn are made up of voxels v. Both the number of blocks in a feature and the number of voxels in a cell is found experimentally and depends largely on the resolution of the 3D scans. For each voxel v within a cell the filter mask [-1,0,1] is applied on its neighbouring voxels in all three dimensions, giving us the 3D gradient vector $\vec{g}$ and its magnitude $\|\vec{g}\|$.

$$(\theta, \phi) = (\cos^{-1}(\frac{g_z}{\sqrt{g_x^2 + g_y^2 + g_z^2}}), \tan^{-1}(g_y, g_x)) \tag{4.3}$$

A weighting w is computed for each voxel based on the gradient magnitude $\|\vec{g}\|$ and the total number of voxels in the cell c, which is used to scale its contribution to that cell's 2D histogram. This is given by the mean value of the voxels within a given three dimensional kernel, indicating the density over this area. By applying this weight, the proposed approach provides accurate estimates also in the presence of noise.

Once these values are established, the voxels within each cell are binned into a 2D histogram h according to their θ and ϕ angles. The value added to a bin is given as the weighted magnitude of the vector $w\|\vec{g}\|$. Finally all cell histograms within a feature h_f are normalised, using the L_2 norm. The resultant histograms can present a way of identifying different types of features and intrinsic properties within an object.

$$h_f \rightarrow \frac{h_f}{\sqrt{\|\vec{g}\|_2^2 + \varepsilon^2}} \tag{4.4}$$

The obtained features are then vectorised and used by the learning mechanism to create a classification model.

$$\vec{x}^{3DVHOG} = \{h_{1,1}, ..., h_{1,\varphi}, ..., h_{\theta,\varphi}\} \tag{4.5}$$

The resultant 2D histograms can be visualised and present a way of identifying different types of features within an object (Figures 4.6 - 4.8 C). Another form of visualisation plots each possible gradient vector within local 3D histograms, showing the most common gradient vectors as stronger (Figures 4.6 - 4.8 A). As can be seen from the various examples, the feature descriptor provides clear differences between the test cases.

The use of voxel weighting smooths the edges of an object cluster ensuring robustness against noisy input data. Due to the local nature of the proposed feature, issues related to the normalization of a mesh are avoided, removing a potentially complex preprocessing step.

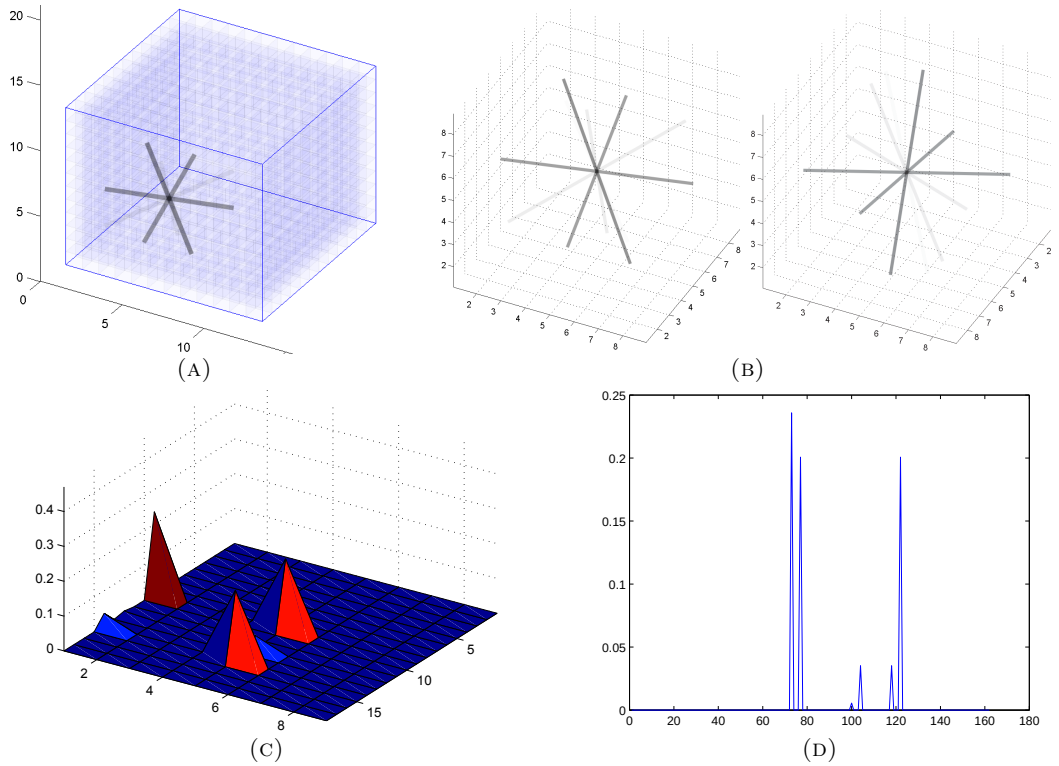


FIGURE 4.6: 3D Voxel HOG feature from a cube wall test case, (A) visualised on its object in 3D, (B) the same 3D representation in two different orientations, (C) as a 2D Histogram and (D) as a 162 dimension feature vector

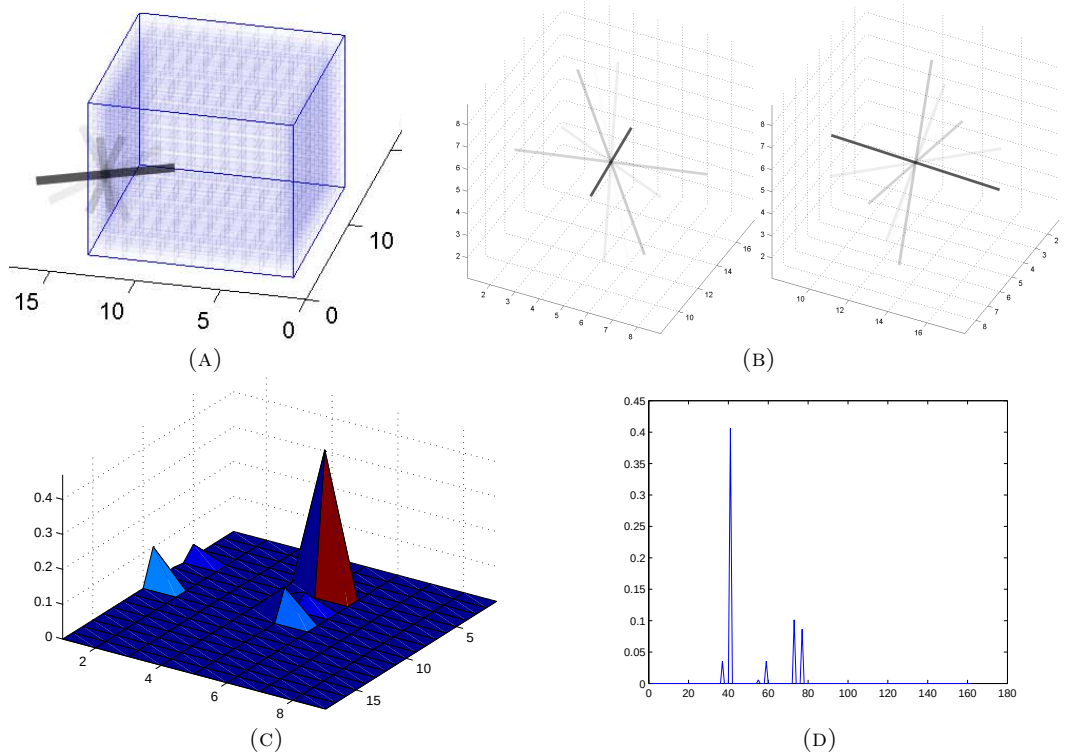


FIGURE 4.7: 3D Voxel HOG feature from a cube edge test case, (A) visualised on its object in 3D, (B) the same 3D representation in two different orientations, (C) as a 2D Histogram and (D) as a 162 dimension feature vector

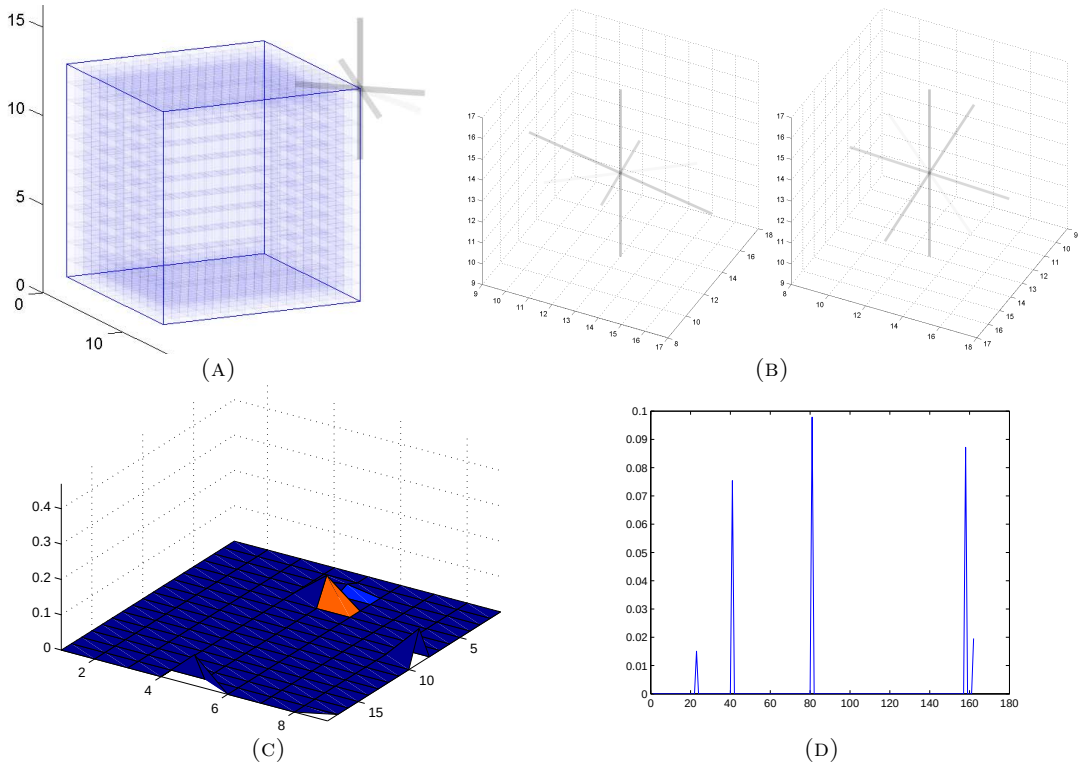


FIGURE 4.8: 3D Voxel HOG feature from a cube corner test case, (A) visualised on its object in 3D, (B) the same 3D representation in two different orientations, (C) as a 2D Histogram and (D) as a 162 dimension feature vector

The pseudo code for the 3D Voxel HOG implementation is outlined below.

```
1. choose Size of Cell and Feature Block
2. FOREACH Voxel v DO
3.     compute Weight w, GradientVector(g⃗),
         Vector Magnitude ‖g⃗‖
   end
4. FOREACH Cell c in Feature Block f DO
5.     create blockHistogram(θ_bins, φ_bins)
6.     FOREACH voxel v in c DO
7.         insert w‖g⃗‖ into blockHistogram(θ, φ)
       end
   end
8. L2Normalize(blockHistogram in Feature)
end
```

These features are used to create a trained model that unknown shape features can be tested against. A binary classification is returned defining the object as either being

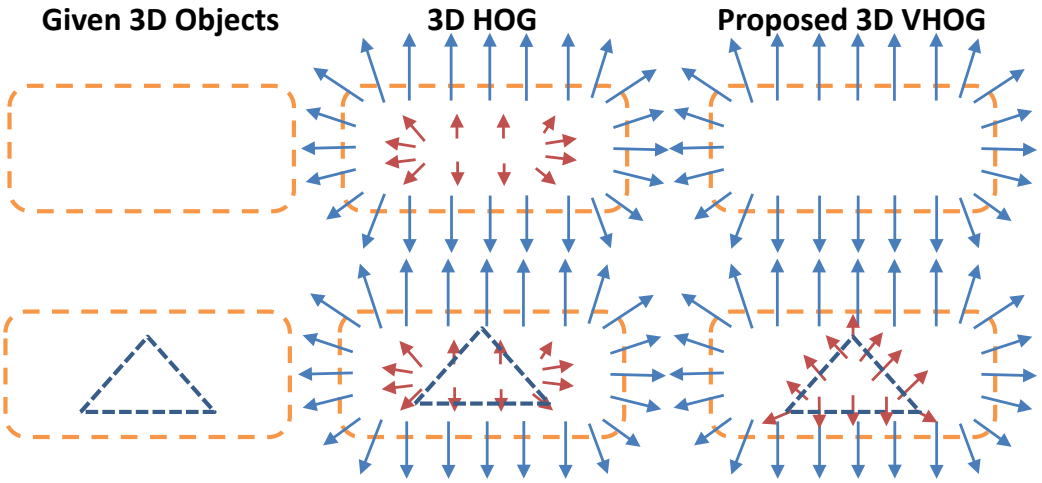


FIGURE 4.9: Example showing the differences of the proposed 3D Voxel HOG features with the 3D HOG [17] indicating that the objects' internal density affects the proposed 3D VHOG descriptor.

hazardous or not.

One of the primary advantages of the proposed 3D VHOG is the consideration of not only the faces of a mesh but also the area within. This ensures that no additional artifacts are created within the data that may lead to false classifications. Furthermore the density of an object can also be taken into consideration. This allows transference of the methodology to other areas such as medical imaging, for example the proposed method could potentially detect defects such as osteoporosis in bone MRI scans which existing methods would not. A visual comparison of the 3D HOG features suggested in [17] against 3D VHOG is shown in figure 4.9 indicating that the proposed method does not introduce erroneous information in the internal areas of an object. Importantly 3D VHOG returns one 2D histogram (visualised in 3D) per cell (Figure 4.6), as apposed to the other methods that provide multiple 1D histograms (visualised in 2D).

Finally, in order to define the 'hazard features' element $e_2 = H$ for the 3D Voxel HOG and the PBF features for the risk score R in (3.1), the obtained outcomes from the classification process are utilised.

$$H^{3DVHOG} = \frac{1}{i}\sum_{j=1}^{i}\left(\frac{\sum_{k=1}^{m} C(j,k)G(j,k)}{\sum_{k=1}^{m} G(j,k)}\right) \tag{4.6}$$

$$H^{PBF} = \frac{1}{i}\sum_{j=1}^{i} C(j) \tag{4.7}$$

where $C(x) \in \{-1, 1\}$ is the normalised classification result from Adaboost for a specific feature x, where positive values indicate the detection of a hazardous feature and $G = \frac{1}{2}(sign(C(x)) + 1)$. Here i represents the number of objects within a scene and m the number of features describing that object.

4.3.4 Robust Kernel

Other descriptors that could be used to identify hazardous objects based on their intrinsic properties (e.g. sharp, pointed) are 3D local shape features such as 3DSIFT [146, 147], 3DHOG [68], 3D Harris [148], FAST 3D [149]. Supervised learning techniques are utilised to classify the objects as hazardous or not but, due to noise of the RGBD acquisition devices and their low resolution, the obtained accuracy is affected significantly. As a result of this, careful attention must be given to the outliers ensuring that the classification accuracy is reliable and remains as high as possible. In the following analysis the Robust Kernel for 3D local descriptors is outlined using 3D Voxel HOG as an example. However this process is applicable to any feature vector without any modifications.

Let $\vec{x}^{3DVHOG}$ be the p-dimensional vector obtained by applying the 3D Voxel HOG (3D VHOG) in an area of a given scene. Based on the work in [150] on robust correlation translation estimation, the L_2-norm is replaced with the dissimilarity measure below:

$$d(\vec{x}_1^{3DVHOG}, \vec{x}_2^{3DVHOG}) = \sum_{c}\{1 - cos(\alpha\pi(\vec{x}_1^{3DVHOG} - \vec{x}_2^{3DVHOG}))\} \tag{4.8}$$

where the values of the corresponding 3D VHOG features $\vec{x}_1^{3DVHOG}$, $\vec{x}_2^{3DVHOG}$ are represented in the range $[0, 1]$. A small value for α results in a function which resembles the L_2-norm. With increasing α the effect of large distances, possibly caused by outliers,

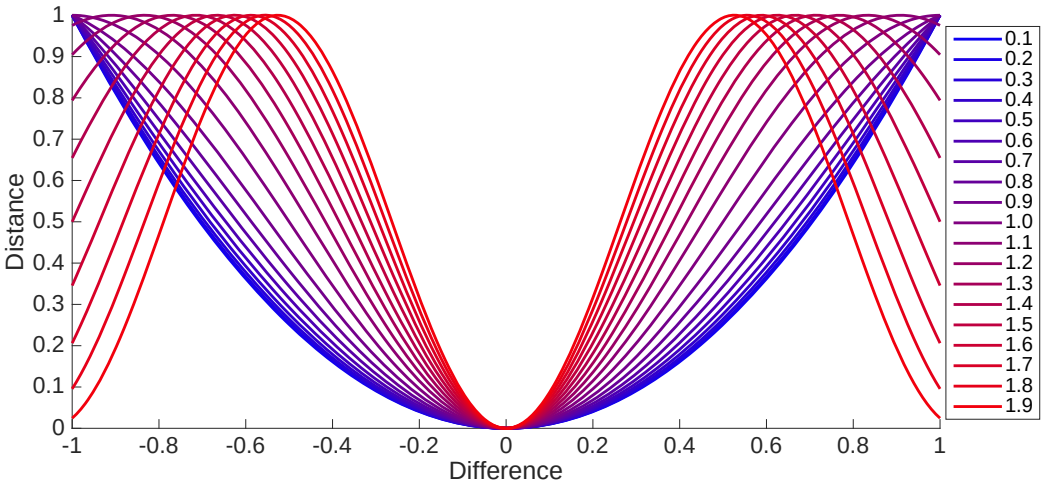


FIGURE 4.10: Illustration of the effect that the α value of the Robust Filter has on a range of distances

is reduced. In general, α represents the frequency of the cosine and is optimized to suppress the values caused by outliers (Figure 4.10).

Based on equation 4.8 and the work in [151], this kernel can be represented using the Euler form of complex numbers. In more detail, the angle values of $\vec{x}^{3DVHOG}$ normalised in $[0, 1]$ are mapped onto the complex representation $\vec{z}^{3DVHOG}$

$$\vec{z}^{3DVHOG} = \frac{1}{\sqrt{2}} e^{i\alpha\pi\vec{x}^{3DVHOG}} \tag{4.9}$$

The values of $\vec{z}^{3DVHOG}$ will be now considered the feature vector used in our learning mechanism. The proposed robust 3D VHOG is a descriptor feature refinement, which aims to reduce the effects of these outliers. The same kernel can be used without any modification by the other descriptors such as 3D SIFT, 3D HOG and 3D Harris.

4.4 Methodology: Learning via Boosting

This part of the proposed framework for risk estimation and scene analysis concerns the classification process which is based on supervised boosting techniques. In this section a novel extension of Adaboost is proposed to handle complex or hyper-complex feature vectors such as those produced by the proposed Robust Kernel for the 3D VHOG descriptor or any other similar one.

4.4.1 Adaboost

Adaboost is a learning technique that creates a non linear classifier to separate data into two groups. Weak classifiers are defined with a final strong classifier being a combination of these. At each iteration the weak classifiers with the lowest error margin are used to define the next in a 'greedy fashion'. Regarding the proposed features in both cases given N training examples $(\vec{x}_1, ..., \vec{x}_N)$, the corresponding labels $(y_1, ..., y_N)$ with $y_i \in \{-1, 1\}$, and an initial distribution of weights $w_1(i)$ a strong classification model $\mathbf{B}(x)$ is obtained based on the weak classifiers β. The weak classifiers are trained over a number of iterations J using the weights' distribution w_j. In each iteration the error ϵ_j is estimated based on the current weights w_j, that are updated before the next iteration.

$$w_{j+1}(i) = w_j(i) \exp(-\alpha_j y_i \beta_j(x_i))/Z_j \tag{4.10}$$

where $\alpha_t = -\frac{1}{2}\log(\epsilon_j/(1-\epsilon_j))$ and $Z_j = 2\sqrt{\epsilon_j(1-\epsilon_j)}$ is a normalization factor. The strong classifier is defined as $\mathbf{B}(x) = sign(f(x))$, where $f(x) = \frac{\vec{\alpha}\cdot\vec{\beta}(x)}{\|\vec{\alpha}\|_1}$.

Regarding the boosting approach, because of the way weak classifiers are selected, a complicated feature problem can be broken down and classified using a sparse classification rule, based on only a few features. This makes computation much faster as only a subset of the features is used. This is essential if the methodology is to be implemented in a real time scenario. Another advantage of this approach is the explicit minimisation of error, whilst implicitly maximising the margin. This ensures the final strong classifier is general avoiding the problems of overfitting.

Another potentially valid boosting technique uses Support Vector Machines(SVM) which also provides a non linear, robust classifier. However SVM tends to have higher computational requirements due to the classifier taking into account all the features in a vector as apposed to just a subset [152].

4.4.2 Complex Adaboost and Hyper Complex Adaboost

In this section Complex and Hyper Complex Adaboost is presented, which implement a modification to the traditional Adaboost utilising complex numbers for use within weak

classifiers suitable for the proposed robust kernel. The motivation for the proposed complex Adaboost comes from the proposed robust descriptor. The descriptor encodes histograms as angular data of the form $z = \cos(a) + j\sin(a)$. In this space, to measure similarity a Hermitian inner product between two descriptors z_1 and z_2 can be defined as $z_1^H z_2$. Although one can replace this with a concatenation of the cosines and sines of the form $x = [\cos(a); \sin(a)]$ and then measure similarity using the familiar inner product $x_1^T x_2$, this implies independence between the elements of the feature vector. This assumption is not always valid, and although commonly accepted, it may lead to a loss of discriminative richness of the vectorial features [153, 154], which can be exploited further by considering the correlation information between the components.

In Adaboost, each weak classifier β_j must determine the optimum threshold per feature dimension that minimises the classification error ε_k, as described in (4.11).

$$\beta_j = \arg\min_{\beta_k \in \mathbf{B}} \varepsilon_k = \sum_{i=1}^{m} D_j(i)[y_i \neq \beta_k(x_i)] \tag{4.11}$$

with D_j being the importance weight for each sample i, with value x_i and label y_i, at each iteration j. D_j is given by

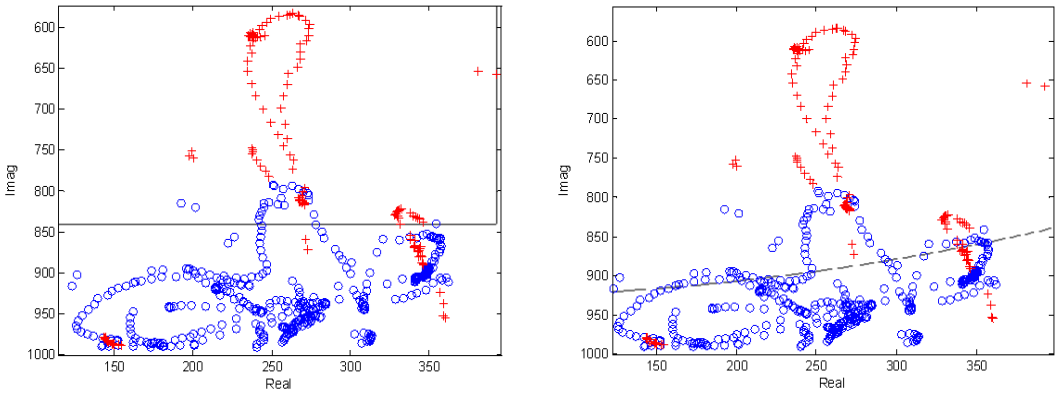
$$D_j(i) = \frac{D_{j-1}(i) \cdot e^{-\alpha_j y_i \beta_{j-1}(x_i)}}{Z_{j-1}} \tag{4.12}$$

where Z_{j-1} is a normalization factor chosen so that D_j is a distribution.

There exists many methods in which this decision can be calculated; one such optimised and fast approach [155] computes cumulative histograms per feature for each of the classes. The histograms allow for the selection of a thresholding bin, chosen to maximise the number of samples of one class whilst minimising the number of the other. The point of minimum error is obtained and for each iteration step of the Adaboost algorithm, the feature with the lowest minimum error is selected as the weak classifier.

This concept forms the foundation of the proposed method. Cumulative histograms per feature are modelled as bi-dimensional distributions allowing for the use of complex

numbers. The use of complex number theory extends the interpretation of a linear one dimensional space into two. Within this space, any given complex number $z = a + bi$ is now represented as a point (a, b). This alters the mathematical meaning and significance of concepts such as minimum and maximum, thus altering the actual definition and implementation of the weak classifiers.



(A) (B)

FIGURE 4.11: Decision border calculated by the first weak classifier on the complex space considering (A) a linear border or (B) a curve one.

As before, a threshold point is obtained which takes into account that the max and min operators have a different interpretation in the complex number space. The threshold is used as a linear decision border by applying the operators to the real and imaginary parts or as a curve border by applying it to the magnitude and angle (Figure 4.11). In the same way the complex number space can also be reinterpreted as polar coordinates rather than cartesian, by using the real and imaginary coordinates as modulus and phase prior to the creation of the bi-dimensional histograms.

With either case it is important to outline the differences that the proposed methodology has to the conventional Adaboost with regards to the real and imaginary parts as independent features. In essence using conventional Adaboost in this way would not respect the complex number nature of the feature source. The relationship between the imaginary and real numbers is not independent but interrelated as a result of the complex number phenomenon. Thus by considering them in isolation that link is lost which leads to a less rich decision as only half of the information is available when the optimisation search is applied.

To preserve this link the optimization search to find the threshold, which provides the

minimum error in the feature space, is extended from one dimension into a two dimensional search. However this increases computational time. To avoid this, an efficient use of feature data is integrated into the methodology requiring fewer iterations. The cumulative distributions are calculated by applying the integral image [156, 157]. Instead of evaluating each possible hypothesis until the optimum is found (leading to the consequential computational repetition of overlapping areas) a cumulative distribution function is precalculated. The application of the integral image technique allows us, in a single pass over the distribution, to efficiently compute a bi-dimensional cumulative distribution function using the following equation:

$$Q(f,r) = Q(f,r-1) + Q(f-1,r) - Q(f-1,f-1) + d(f,r) \tag{4.13}$$

where d is the original distribution function, modelled as a histogram. Q is the cumulative integral image and f and r are the column and row indexes, respectively.

In a similar manner that complex numbers extend the feature space to a two dimensional space, quaternions extend it to a four dimensional space (and to three dimensions in case of pure quaternions). As such the proposed methodology is extendable to higher numbers of dimensions, importantly without assuming independence between the values of these vectors and therefore without losing any of the relational information.

To allow for this, and in the case of quaternions, the optimisation search step must be done in a four dimensional space to find the decision threshold. By replacing the integral image with a multidimensional extension of the integral image [158, 159], the required four dimensional cumulative histogram can be efficiently calculated and the threshold extracted. Therefore (4.13) is transformed to:

$$Q_{DIM} = \sum_{p \in \{0,1\}^{dim}} (-1)^{dim - \|p\|_1} Q(x^p) \tag{4.14}$$

where dim is the image dimension, Q is the bi-dimensional integral image of the histogram h, and x^p represents the multidimensional rectangle $[x_0, x_1]$ to be evaluated at each position.

FIGURE 4.12: Subset of the objects in the 3D Risk Scenes (3DRS) dataset.

4.5 Results

4.5.1 Experiment Environment

4.5.1.1 3D Risk Scenes Dataset

To test the proposed hazard feature detection methodologies the 3D Risk Scenes (3DRS) dataset is utilised. Using the Microsoft's Kinect and Kinect Fusion [4], 27 real objects were captured (Figure 4.12).

Of the 27 objects captured, 12 are classified as hazardous with the remaining 15 safe. These include everyday tools and objects commonly found around the home such as knives, irons, balls, cutlery, mugs, bowls, bottles, computer equipment, scissors and vases. All the objects in the dataset were run through the same preprocessing techniques as in Section 3.3.2, returning a voxel volume containing individual objects from the dataset per volume. For all cases a voxel volume representation is returned with a resolution of 256x256x256 voxels, representing an approximate volume of $50cm^3$. Any lower resolution and shape information about the object would be lost. The returned 3D reconstruction of a scene from Kinect Fusion has some preliminary smoothing and hole filling techniques applied, and therefore any higher resolution would not affect significantly the overall performance. The resolution also has a direct impact on computation time for each stage and as such this represents a reasonable trade off for processing time against object detail.

FIGURE 4.13: Example images of participant expressions at highest intensity (happiness, disgust, anger, surprise, fear and sadness) from BU-3DFE dataset [12].

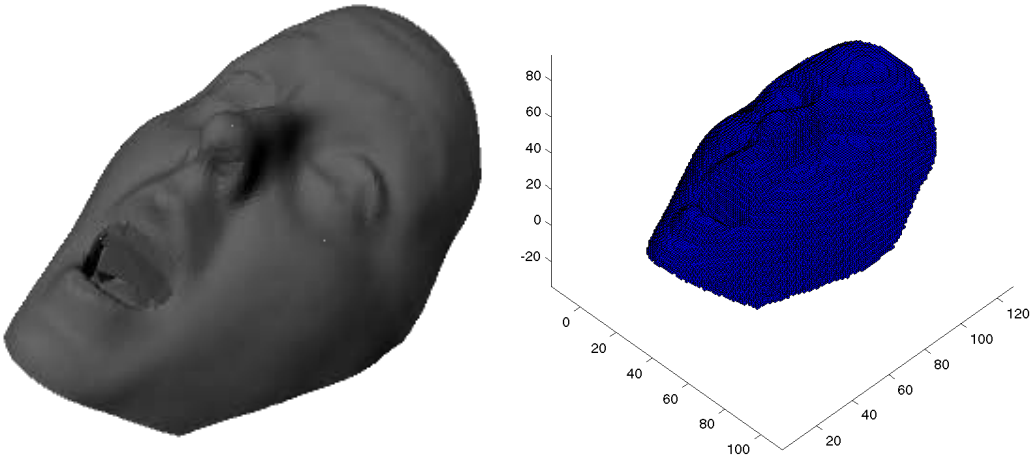


FIGURE 4.14: Example input mesh object of participant face and resultant voxel volume after preprocessing.

4.5.1.2 BU-3DFE Dataset

To further test the effectiveness of the 3D VHOG feature and the robust kernel, the method was applied to features designed for another popular area of computer vision. Utilising the BU-3DFE dataset [12], the common multi-class expression recognition task was performed, highlighting the versatility and applicability for the methodologies within other areas of computer vision research. BU-3DFE comprises 100 participants (44 male and 56 female) within a range of ages and with a number of differing ethnic/racial ancestries. Each participant performed seven different expressions and, with the exception of the neutral expression, each were performed with four levels of intensity (Figure 4.13). For each of these expressions a mesh model of the face is captured providing the input to the outlined preprocessing stages. As with the 3DRS objects, a voxel volume of the face model is returned with a resolution of 256 cubic voxels (Figure 4.14).

4.5.2 Hazard Feature Evaluation

To evaluate the proposed Physics Behaviour Feature (PBF), analysis was conducted on the 27 objects from the 3DRS dataset. Once preprocessed, each object and its resultant bounding shape information was used to perform physics simulations. In order to improve the accuracy of the simulations customised bounding shapes that best suit the objects were used and mass information supplied for each object from the 3DRS dataset.

The outputted data from the simulation environment is a position and rotation of each object at each frame. The x, y and z values represent the location in 3D space of the object (or its rotation). Using this data the velocity or the angular velocity can be calculated. To create the final feature vector, differing combinations of these components were utilized. Several features were investigated and evaluations were carried out to establish which one is the most suitable. Table 4.1 outlines the feature vector combinations carried out and the result F1 score achieved when using Adaboost. Various combinations of strength of force applied, number of forces, and length and source of the feature vector were investigated. For the ground truth an object is defined as either dangerous or not, as such the trained model returns the same classification for each tested object. Testing was carried out in a 'leave-one out' fashion for each of the feature combinations.

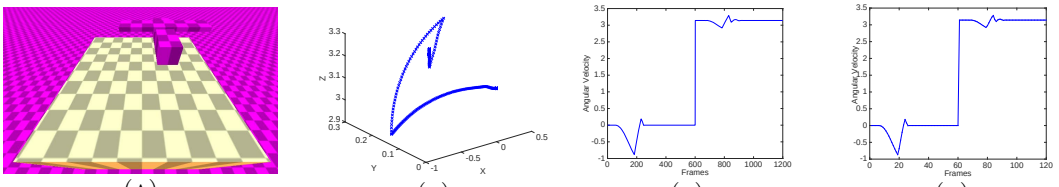
In addition to these tests a number of complex and hyper complex variants were also evaluated. A pure quaternion representation was considered where the use of the x, y and z values make up the imaginary components. Experimentation was also carried out by reducing the initial data down to two dimensions combined in a complex representation. In all the evaluated cases, the features were tested with and without the proposed complex (or hyper-complex) representation. Both of these complex forms compliment the use of the Complex and Hyper Complex Adaboost, allowing the exploitation of the relationships between the dimensions of the data to be taken into account. In this case it was found that the most suitable form was utilising just the x component of the angular velocity.

A visualisation of this feature definition process can be seen in Figure 4.15-4.16 for two different objects. Subfigure (A) shows the collision shapes in the simulation, (B) the 3 components (x,y,z) of the angular velocity plotted over time, (C) the dimensionality

TABLE 4.1: Results of the Physics Behaviour Feature (PBF): combining different number of forces, strengths, axis and length from simulation data captured from the 3DRS dataset

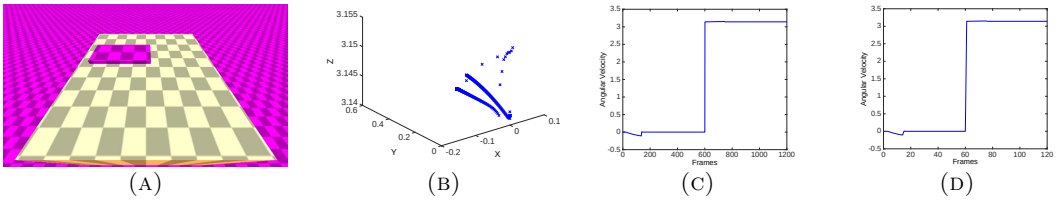
No. Forces	Strength	Axis	Length	Data	Result(F1)
1	Weak	X	30	Rot Direct	0.667
1	Weak	X	60	Rot Direct	0.000
1	Strong	Z	60	Rot Direct	0.417
1	Strong	X,Z	120	Rot Direct	0.250
1	Weak	X	120	Rot Direct	0.200
1	Weak	X,Y,Z	180	Rot Direct	0.200
4	Strong	X	240	Rot Direct	0.690
4	Strong	X,Z	480	Rot Direct	0.435
4	Weak	X,Z	480	Rot Direct	0.400
1	Weak	X	30	Ang Velocity	0.207
1	Weak	X	60	Ang Velocity	0.480
1	Strong	Z	60	Ang Velocity	0.000
1	Strong	X,Z	120	Ang Velocity	0.519
1	Weak	X	120	Ang Velocity	0.421
1	Weak	X,Y,Z	180	Ang Velocity	0.125
4	Strong	X	240	Ang Velocity	0.640
4	Strong	X,Z	480	Ang Velocity	0.538
4	Weak	X,Z	480	Ang Velocity	0.118

reduction and (D) the final feature vector after down-sampling. With this defined variation of the feature descriptor further comparative testing can be carried out against other feature vectors. Table 4.2 outlines this and the results are discussed below.



(A) (B) (C) (D)

FIGURE 4.15: Physics Behaviour Feature extraction. (A) Simulation, (B - C) before and after the dimensionality reduction and (D) after down-sampling.



(A) (B) (C) (D)

FIGURE 4.16: Physics Behaviour Feature extraction. (A) Simulation, (B - C) before and after the dimensionality reduction and (D) after down-sampling.

To test the effectiveness of the 3D VHOG feature descriptor and the PBF, the same 27 objects from the 3DRS dataset were utilised. A range of other 3D shape descriptors were used to provide a comparison. The 3D HOG is based on the work in [17], the 3D SIFT implementation based on the papers [146, 147], the 3D Harris implementation considers the work in [148] and finally the FAST 3D implementation is based on the work in [149].

Most of the tested descriptors operate on local areas of the voxel volume thus the ground truth for each of these blocks or feature spaces is defined. All descriptors were trained with the same training set using Adaboost [33]. For testing the 'leave-one out' protocol was used and a set number of iterations (500) was specified to create the models. This number was found experimentally to produce the best overall classification models for the dataset. In some cases convergence would be reached sooner. Regarding the values for block and cell size, these were set to 2 cubic cells and 16 cubic voxels respectively.

Table 4.2 outlines the results of each 3D feature descriptor on the 3DRS dataset. The results are broken down into various measures: Precision is the fraction of retrieved instances that are relevant, defined as the true positive rate divided by the number of correctly identified classifications. Sensitivity is the fraction of relevant instances that are retrieved, defined as true positive rate divided by the number of positive results that should have been classified. Both precision and sensitivity are therefore based on an understanding and measure of relevance. Accuracy is a description of systematic errors, or a measure of statistical bias. Finally the F1 Score is another measure of accuracy that uses precision and sensitivity to compute its score.

It can be seen that many of the well known feature descriptors are applicable to this task. However 3D Harris and FAST 3D both performed poorly, this is in part down to a lack of convergence when training the model. Additionally a tendency to over fit the model was present and as such does not provide a consistent enough description of this local phenomena, possibly due to small variations in the voxels or due to the voxel resolution of the scene.

When compared with other features, PBF shows promising results in the detection and classification of objects in this approach. The formation of the feature vector has a direct influence on the types of objects that are well classified. This property of the feature could potentially be exploited to classify other aspects of an object. A combination of the proposed physics (PBF) and the shape (3D VHOG) was devised. To ensure the safest results the two features were fused using an 'OR' operator on an object's classification as hazardous. If either PBF *or* 3D VHOG returns a result of hazardous then that object is deemed unsafe. This combination of features allows analysis of an object cluster on both a local level (3D VHOG) but also at an overall shape level (PBF). This combined

TABLE 4.2: Comparison of proposed methodologies versus existing 3D feature methods on the 3DRS dataset objects.

Feature	F1	Sensitivity	Precision	Accuracy
3D HOG	0.699	0.750	0.600	0.667
3D Voxel HOG	0.714	0.833	0.625	0.704
3D SIFT	0.545	0.500	0.600	0.630
3D Harris	0.267	0.167	0.667	0.593
FAST 3D	0.000	0.000	1.000	0.556
PBF	0.690	0.833	0.588	0.667
PBF+3D VHOG	**0.750**	1.000	0.600	0.704

TABLE 4.3: Comparison of 3D feature methods with the addition of the robust kernel on 3DRS dataset objects.

Feature	F1	Sensitivity	Precision	Accuracy
3D HOG	**0.699**	0.750	0.600	0.667
Robust 3D HOG	0.686	1.000	0.522	0.593
3D Voxel HOG	0.714	0.833	0.625	0.704
Robust 3D Voxel HOG	**0.769**	0.833	0.714	0.778
3D SIFT	0.545	0.500	0.600	0.630
Robust 3D SIFT	**0.571**	0.667	0.500	0.566
3D Harris	0.267	0.167	0.667	0.593
Robust 3D Harris	**0.353**	0.250	0.600	0.593
FAST 3D	0.000	0.000	1.000	0.556
Robust FAST 3D	**0.261**	0.250	0.273	0.370
PBF	0.690	0.833	0.588	0.667
Robust PBF	**0.727**	0.667	0.800	0.778
PBF+3D VHOG	0.750	1.000	0.600	0.704
Robust PBF+3D VHOG	**0.828**	1.000	0.706	0.815
Average 3D	0.523	0.583	0.668	0.645
Average Robust 3D	**0.599**	0.666	0.587	0.641

descriptor results in an overall improvement as shown in Table 4.2 indicating that their fusion increases accuracy when recognising hazardous and safe objects.

4.5.3 Robust Filter Evaluation

To evaluate the effectiveness of the proposed robust kernel evaluations were conducted using the 3DRS dataset. The feature vectors for all of the above descriptors had the kernel applied and new models trained with Adaboost using the same parameters as before.

From the average results obtained, the overall F1 score was improved by 7.57% indicating the proposed robust kernel has strong potential for use with most of the well-known 3D descriptors. These results clearly demonstrate improvements can be seen in the F1 score

TABLE 4.4: Comparison of 3D VHOG and other feature methods, with and without the addition of the robust kernel, on the BU-3DFE dataset.

Feature	Standard	Filtered
3D Voxel HOG	69.17	**71.83**
FAST Image	49.50	**51.67**
2D Texture Feature	67.50	**70.00**
Average 3D	62.06	**64.50**

and, in most cases, the sensitivity, on a wide range of 3D descriptors using the proposed filter, providing more accurate and robust classifications.

Utilising the BU-3DFE dataset [12] further tests were conducted on the effectiveness of the Robust Kernel and the diverse applications of the 3D VHOG feature descriptor. The common multiclass expression recognition task was performed, in which the goal is to train a set of models capable of categorising facial expressions into six different emotions. For each of the 100 participants, the highest intensity mesh models for each of the six emotions were used.

As before a number of feature descriptors were used in the machine learning process including 3D VHOG, FAST Image (an amalgamation of FAST 3D and gaussian image) and a 2D feature descriptor based on the texture images. The reasoning behind the feature choice was to give a range of descriptors that are of differing structures and methodologies which better illustrate that the Robust Kernel is applicable to many types of feature vector. Training is done utilising an 80/20 split of the data. Multi-Adaboost is applied using the one-against-all approach by constructing binary classifiers for each expression class. In order to obtain the final classification, the individual results are combined using a majority vote. A comparison is drawn against their effectiveness with and without the Robust Kernel and results are given as the percentage of expressions correctly classified into the six different emotion expressions.

The results show a higher recognition rate for each feature type when using the Robust Kernel. During the experimentation process the percentage increase that the robust kernel yielded ranged from 0.6% up to 6.01%.

4.5.4 Complex and Hyper Complex Adaboost Evaluation

To evaluate the advantages of the proposed Complex Adaboost the complex 3D feature vectors obtained after using the Robust Kernel were compared with the classic Adaboost

TABLE 4.5: Complex Adaboost vs standard Adaboost, training times and iterations comparison on the 3DRS dataset objects.

	Standard Adaboost		Complex Adaboost	
Feature	Time(s)	Avg #Iter.	Time(s)	Avg #Iter.
3D VHOG	348.57	103.96	9.08	46.96
Robust 3D VHOG	651.46	40.67	45.15	14.82
3D Sift	855.16	72.92	19.95	72.92
Robust 3D Sift	1603.66	61.74	43.77	78.96
3D Harris	2261.00	500	46.268	500
Robust 3D Harris	4576.38	500	106.71	500
Fast 3D	2351.40	500	52.33	500
Robust Fast 3D	15959.70	500	93.88	500
PBF	162.56	4.41	1.44	9.26
Robust PBF	1781.7	4.15	4.04	6.98
Average 3D	1195.74	236.26	**25.81**	**225.83**
Average Robust 3D	4914.58	221.31	**58.71**	**220.15**

TABLE 4.6: Diagnostic testing of results against existing 3D feature methods using Complex Adaboost on the 3DRS dataset objects.

Feature	F1	Sensitivity	Precision	Accuracy
Average Adaboost	**0.544**	0.516	0.701	0.659
Average Complex Adaboost	0.512	0.467	0.686	0.630

in terms of complexity. A comparison is given in terms of the training time and the number of iterations required. As before the maximum number of training iterations was specified to 500. Testing was carried out on an i7-4870 2.5GHz PC with 16GB RAM running Windows 8.

The results in Table 4.5 were derived from the average results from the 27 generated models for each object for each descriptor type. The iterations were limited to 500, thus results which reached this number of iterations did not converge. We can see that computational speed gain is considerable with similar numbers of iterations being completed within a fraction of the time needed with conventional Adaboost.

Diagnostic testing of the generated results show slightly lower F1 scores on average (Table 4.6), demonstrating that where time is a consideration the proposed complex version of Adaboost is very effective.

To outline the advantages of Hyper Complex Adaboost, experiments were conducted on a three dimensional permutation of the PBF. Sixteen different feature vector combinations, utilising all three axis of either the angular velocity, rotational velocity or position, were analysed using both Adaboost and the proposed Hyper Complex Adaboost. The feature vectors were either concatenated vectors of all the data or Hyper Complex variants where

TABLE 4.7: Hyper Complex (HC) Adaboost vs standard Adaboost accuracy evaluation on the 3DRS dataset objects.

	F1	Sensitivity	Precision	Accuracy
3 Axis Feature w/ Adaboost	0.293	0.276	0.389	0.488
3 Axis Feature w/ HC Adaboost	0.292	0.318	0.292	0.456
HC Feature w/Adaboost	0.108	0.073	0.210	0.472
HC Feature w/HC Adaboost	**0.348**	**0.365**	**0.438**	**0.537**

TABLE 4.8: Risk Score of individual objects calculated using PBF+VHOG feature.

Object	Bal	Bot	Bow	Con	Fra	Ham	Hed	Ir	Ir2
Hazard Score	0.06	0.00	0.03	0.16	0.93	0.78	0.32	0.77	1.00
Object	Kn	Kn2	Kn3	Kn4	Lp	Lp2	Lap	Mse	Mug
Hazard Score	0.86	0.86	0.82	0.82	0.10	0.22	0.76	0.21	0.25
Object	Pnc	Pno	Pen	Rub	Slt	Sc	Sc2	Scr	Spt
Hazard Score	0.02	0.78	0.88	1.00	0.93	0.76	0.76	0.92	0.79

the three axis made up the imaginary components of the hyper complex number. The average results for the 16 experiments is shown in Table 4.7. As can be expected the results of the hyper complex variant of the feature vector with the standard Adaboost has the lowest average results. Utilising the hyper complex feature vector with the proposed Hyper Complex Adaboost achieved the highest rate of accuracy overall. The results are comparatively low and as such the use of the hyper complex feature vector in the final PBF+3DVHOG feature was detrimental to performance. However these results illustrate the advantages of the use of hyper complex features and the proposed Hyper Complex Adaboost.

4.5.5 Risk Score

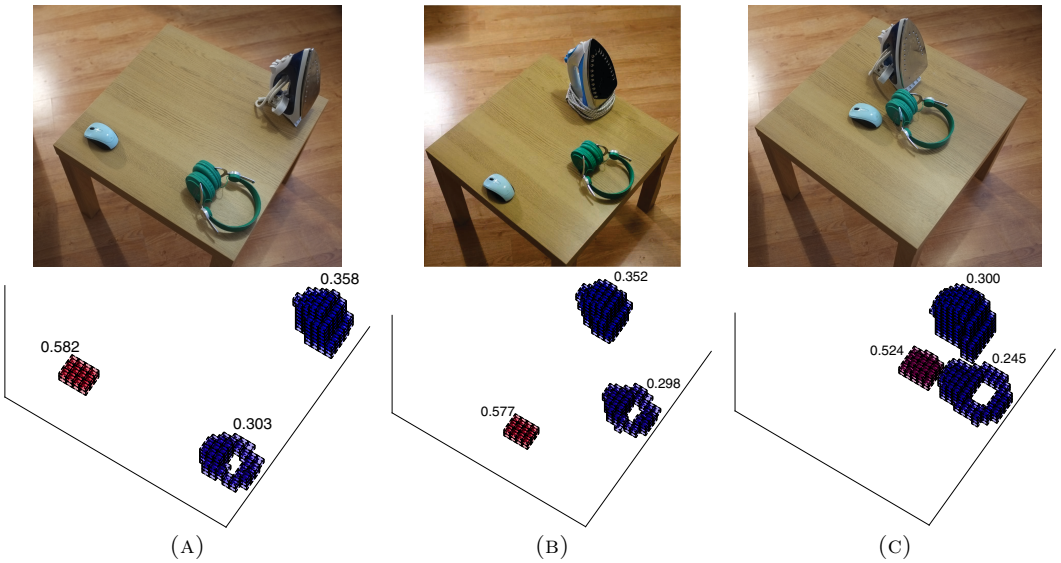
Table 4.8 outlines the hazard scores of each object of the 3DRS dataset according to the PBF+3DVHOG feature descriptor. It can be seen that in most cases the risk score is high for objects that demonstrate some kind of risk e.g the four types of knives, the irons, hammer and the two sets of scissors. Equally less hazardous items are scored low; the ball, bowl, mug. However there are cases where the descriptor has been over sensitive; the rubik's cube and laptop being examples of this. In the given scenarios it is important for the descriptor to be over sensitive to risk so as to ensure that no hazards are overlooked.

Using the updated Risk Estimation framework equation 4.1, a risk score can be derived using both stability and hazard features estimation. As stability estimation is dependant

TABLE 4.9: Risk score per scene. Using PBF+VHOG feature and stability estimation

Scene	1	2	3	4	5	6	7	8
Lv1	0.271	0.414	0.395	0.356	0.632	0.923	0.235	0.535
Lv2	0.268	0.409	0.392	0.344	0.623	0.911	0.231	0.533
Lv3	0.253	0.357	0.374	0.317	0.578	0.888	0.214	0.481
Scene	9	10	11	12	13	14	15	16
Lv1	0.250	0.368	0.226	0.504	0.240	0.509	0.650	1.00
Lv2	0.246	0.349	0.224	0.482	0.239	0.432	0.573	0.983
Lv3	0.229	0.328	0.214	0.481	0.229	0.312	0.452	0.704

on analysis within a scene, the 48 scenes from the 3DRS dataset are used to generate these scores, with hazard feature estimation derived using the Robust PBF+3D VHOG feature descriptor. The results for these scores are shown in Table 4.9. The ground truth for the unsafe objects is provided within the 3DRS dataset where each object is labeled as safe or not. Total risk is defined as the weighted sum of the Hazard and Instability scores with $w_S = w_H = 0.5$ for all the scenes. Figure 4.17 demonstrates the compound risk score for a sample scene. This compares with the instability only evaluation of the scene is Section 3, Figure 3.19. As the weighting for each risk element is equal in this case, the effect is that the risk scores are smoothed out over the different iterations. With the adjustment of these weights a system can be designed to better illustrate relevant risk in a given environment.



(A) (B) (C)

FIGURE 4.17: Illustration of compound instability and hazard feature per iteration of an example scene.

4.5.6 Conclusion

Within this chapter the concept of risk analysis is analysed. Risk estimation is considered in the form of hazard feature analysis, with a view to identifying low level properties of unseen objects such as sharp blades or points. Using the introduced 3D Voxel HOG and Physics Behaviour Feature, testing is carried out on the objects within the 3DRS dataset. A robust kernel for 3D descriptors is introduced which aims to reduce the effect that outliers have on training data, helping to create more robust models for classification tasks. Experiments were performed showing that the proposed approaches have the potential to accurately measure risk in scenes providing good estimates. A complex and hyper complex version of Adaboost is suggested that can exploit the correlation between the real and imaginary elements of complex descriptors with lower complexity, resulting in faster and few training iterations.

4.6 Discussion

With the experimentation performed using the proposed methodologies within this chapter it can be seen that good progress has been made in the identification of risk related properties of objects. With the addition of the stability estimation techniques defined in the previous chapter, a more rounded view of risk in a scene starts to take shape. These components provide a broad overview of risk which is designed to be an general as possible, however a system capable of extracting further object properties, such as weight, would provide a better ans equally general approach to object risk.

Currently this concept of risk is static and at present fails to take into account that none of the highlighted risks are valid if there is never going to be any interaction between the end user and the defined risk. For example, a classified hazardous object in a potentially unstable position on a table is entirely safe if no one ever goes near that area of the scene.

As such a human element is required in the quantification of risk in a scene. Identification of the movement habits of users and how they may react to a given risk would help to provide a more complete picture of risk going forward. Thus with the ability to identify a hazard and predict these interaction trends, it becomes possible to develop a system

which could draw attention to these problems or even correct them before a risk becomes an accident.

Chapter 5

Human Behaviour and the Effect on Risk

5.1 Introduction

As seen in the previous two chapters, it is possible to evaluate risk and return a contextual risk score for a given environment. So far this has focused on the risks found within the environment, measuring how hazardous objects in a given scene are and if the stability of their position poses a potential risk. However, these risk measures do not take into account how human interaction with an environment can be an important factor in deciding if a potential risk is a genuine hazard. Take the example of a knife balanced at the edge of a table. This is only a risk if there is a likelihood that someone is going to interact with it. This presents the need to evaluate a scene based on human interaction. To this end a mechanism is required to predict where people are likely to be in a scene and how they navigate from one part of the environment to another. The provision of a framework which can analyse these properties of an environment allows for smart enabled homes or domestic robots to go into an unknown setting and build up a view of the potential risks. This is important in an applicable system, as complex and long winded configuration steps would limit the potential applications.

As discussed in Section 2.3.1, many techniques exist in the context of simulating pedestrian and crowd behaviour. This has led to extensive research into the movement of

people around environments, having applications in a large range of industries including pedestrian facility suitability and capacity [95], computer graphics and gaming [10], the social sciences [96] and engineering [97]. These methods provide the ability to help evaluate and simulate the effect an environment might have on a crowd and vice versa. Work in this area tends to focus on the simulation of large numbers of agents, given that often the object of the research is to view the impact groups of people have in a given situation. This can be in the context of an evacuation plan or the design of a building, ensuring the highest flow of traffic is achieved.

For the concept of risk evaluation, the accurate simulation of individual agents in the environment is crucial. The ability to accurately emulate human path finding behaviour allows the generation of likely paths which could be used to traverse from one part of the environment to another. Using a virtual representation of a given environment, and by defining key points in the room, a heatmap can be built up representing the areas which humans are most likely to interact with. Key points can include entrances and exits and points of interest, such as areas to sit or objects to interact with. As an example, a seating area in a room will have a higher rate of interaction than the area around a blank section of wall. Using this idea of interaction likelihood, the hazardousness of detected risks present in the environment change based on the frequency of predicted human presence.

Conversely this works both ways; as humans constantly evaluate the environment they are in, the discovery of a hazardous object may well change the way they interact with that environment. The emulation of this decision making behaviour also provides important insight, potentially changing the risk of other areas of a room based on this change in behaviour.

The realism of the simulation algorithms is very important and methods by which to judge the effectiveness of these algorithms is a developing area of research. However one of the most prominent issues with crowd and pedestrian simulation research is the lack of a simple and suitable form of comparison between different simulation and modelling approaches. This often means that a given methodology is developed and evaluated for a specific purpose, often meaning its wider abilities are left unconfirmed. This task is made more difficult as the developed approaches cover a huge range of applications, where evaluation techniques for one are not always applicable to the others.

Generally the evaluation techniques utilised can be split into qualitative and quantitative measures. The former including assessments made by experts in the field or context of the intended application, as well as category based rating systems designed to define the capabilities of an algorithm (such as emergent behaviours). These assess whether the simulation *looks* natural and that the agents within the simulation are not acting in an unusual fashion.

A number of quantitative measures have been suggested which include but are not limited to: speed, pedestrian density, number of steps taken to destination and duration. Additionally evaluation frameworks have been suggested before, deducing various metrics based on a simulation in an effort to rate simulation algorithms or tune parameters. Often these frameworks evaluate a simulation based on their deviation from source data. This is based on the assumption that a good simulation closely matches the captured source data. For example does simulated agent A's track match pedestrian A's track in the source data, and how well does it do so. However humans moving through the same environments on a regular basis will look similar but have slightly different properties.

Many of these evaluation frameworks have merit in their given context. However to make a comparison often a number of requirements are imposed on their source data. Defined tracks for pedestrians in source data is commonly required, this introduces issues to the data collection process pertaining to cost, time, ethics and suitability for large outdoor environments. Additionally the focus of comparison is often given to a statistical analysis of the simulations, specifically on individual agents rather that of the simulation as a whole. This means the way the simulation appears is often overlooked. In some cases the visual realism is addressed however it is often not validated against some kind of human analysis.

The contributions in this chapter are two-fold; firstly the notion of environmental risk maps is approached using a novel behaviour modelling algorithm, based on the presence of risk, providing both evaluation of human interaction with an environment and the subsequent effect the environment has on those within it. Secondly a novel framework is presented providing a method of comparison in which human behaviour algorithms can be evaluated based on how realistic the simulation looks using only un-annotated video footage. Comparison is done using a sample video and simulated footage created using visualisation techniques. Evaluation metrics are proposed that compare the two

video sequences using Human Visual System (HVS) features, which aim to emulate the way humans interpret video sequences.

5.2 Related Work

Within this section a brief overview is given on directly relevant work which this chapter seeks to compare with, improve upon or make direct use of.

Zheng et al [11, 16], analyse a scene based on the probability of an object being dislodged using disturbance fields. By modeling human actions and natural events such as earthquakes or wind effects, the probability of objects falling can be calculated. However, although the probability of where a human is likely to be is computed, the concept of if the human is aware of the risk is not taken into consideration. For example if a human sees a potential risk they are likely to change the way they interact with that environment thus reducing the chance of interacting with that hazard.

Wang et al [130] present the SV-DHDP model in which trending paths in source data can be combined to generate overall path patterns. Using these patterns, visualisations are produced allowing for qualitative analysis. Additionally an inference based similarity metric is also proposed, allowing for the comparison of extracted path patterns from differing data sources. However analysis is done on defined paths for source and test data which requires complex post/preprocessing techniques or data captured in a specific format which, as demonstrated in [95], can cause unnecessary inaccuracies.

Charalambous et al [125] create an analysis tool which compares simulated and reference data to find outlying behaviour. Two processes are suggested. Firstly outlier detection, which takes a set of data and searches for unusual agent behaviour. Secondly novelty detection, in which simulated data is compared to reference material to find and describe trends or actions that differ. Results of the analysis are presented to the user in a number of forms that aim to highlight specific agents that are acting erroneously or where general areas of inconsistency appear. The fundamental issue with the process is the need for the reference data to be very similar to the simulated, indeed both forms of data require predefined tracks for all agents/pedestrians. Additionally as the comparison made is purely a data driven approach, the concept of something *looking* similar is not addressed. For example if an agent's path in a simulated crowd differs from a real world example, it

does not mean the simulated behaviour doesn't *look* real. Additionally analysis is given on an individual agent basis with no global similarity measure given.

Guy et al [126] uses an entropy score to compare simulated to captured real world data. The entropy score provides a measure of difference for a simulation and produces a technique that allows tuning of the simulation so as to closely match that of the real world example. This statistical analysis again relies on the need for position information from both the simulation and comparable real world examples. Additionally a number of assumptions are made, the most noticeable being that the simulation algorithm is not systemically more accurate for some agents within a crowd than for others. However this is not always the case, as there will always be aspects of a simulation that are more accurate than others. The work in [125] demonstrates this.

Finally, Weber's [160] work and that of [161, 162] on how the Human Visual System processes light is utilised within this work to make the proposed similarity metrics more closely resemble the functionality of the human eye.

5.3 Methodology

5.3.1 Risk Estimation Framework

The Risk Estimation framework produces a risk score from any measurable element of risk. Included so far have been the measures of stability S and hazard features H. An additional measure of risk is now introduced in the form of environmental risk maps E. This measure defines the likelihood that a human will come into contact and see a given risk. The value of E can range from zero to one where zero is a low chance of interaction and one high. Again a weighting w_E is specified and represents the contribution that the risk element E has to the final risk score. The sum of the weightings for all included risk elements is equal to one. With this additional element the Risk Estimation framework (3.1) is redefined, such that $n = 3$ where $w_1e_1 = w_SS$, $w_2e_2 = w_HH$ and $w_3e_3 = w_EE$.

$$R = w_SS + w_HH + w_EE \tag{5.1}$$

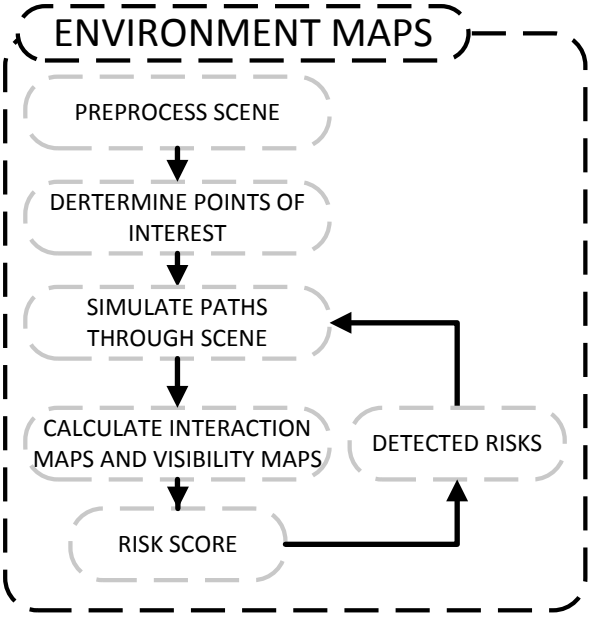


FIGURE 5.1: Overview of the environment maps process and the resultant risk score generated.

5.3.2 Environmental Risk Maps

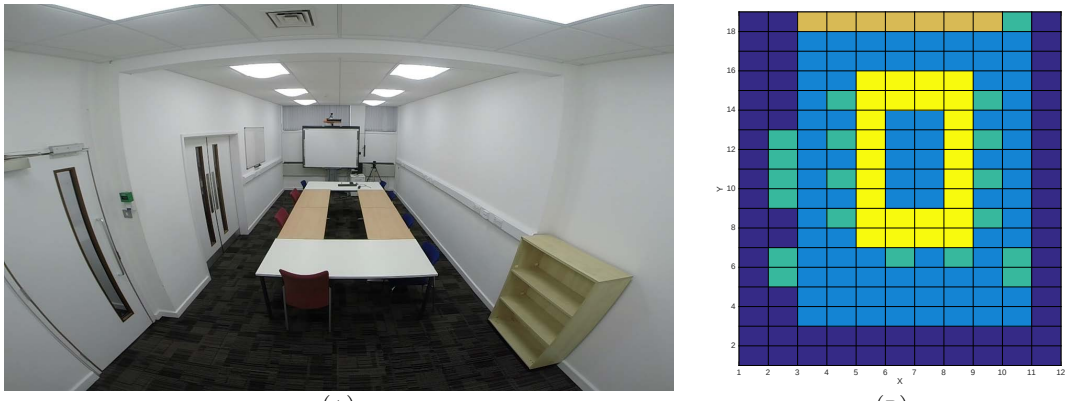
To simulate interaction with a scene it is important to consider a number of aspects. Firstly and most notability, the expected paths that humans would take through a scene. In the example of a kitchen; this would include paths through the room via entrances and exits, as well as paths to and from facilities within the room such as sinks, cookers, fridges and other likely destinations. Second to this is the concept of visibility, how much of the scene is visible to the average human as they navigate the environment. For example if someone walks past a table, they can likely see on top of the table but not what is below or behind. Finally the concept of redirection on account of a hazard. Here simulations take into account that a human walking though an environment may change their initial planned path on account of a discovered risk. As such other areas within the environment would have an increased likelihood of interaction.

An overview of the process is given in Figure 5.1. After the scene is preprocessed and a map and points of interest has been generated for the environment. Simulation algorithms are utilised to define the likely paths through the scene. From this simulated data interaction and visibility maps are generated and risk scores defined. In the presence of a detected risk, simulations can be rerun to take this into account allowing the continuous update of the risk scores to reflect the current state of the scene.

5.3.2.1 Interaction Maps

By tracking a human's movement through an environment over time a picture can be created describing which areas are used more frequently than others. This method requires time and is specific to each individual environment. However similar results can be quickly produced through the use of simulation methods. As such the emulation of human behaviour with regards to the navigation and interaction with domestic environments forms the foundation on which the environment maps and subsequent risk estimation techniques are based. A detailed explanation of the simulation techniques utilised is given in Section 5.3.3.

Initially to allow simulation of an environment to be run, mapping of that environment is required. This can be achieved through a number of methods [4, 163–165] with the detection of entrances through use of existing techniques [166]. By creating a low resolution two dimensional map of an environment, with labels defining points of interest, multiple simulations can be run to replace the long term monitoring techniques. For this work, 2D low resolution maps of the environment are used, obtained either using the methodology highlighted in Section 5.3.4.1, or created manually. The entrance/exit and points of interest have been labeled manually.



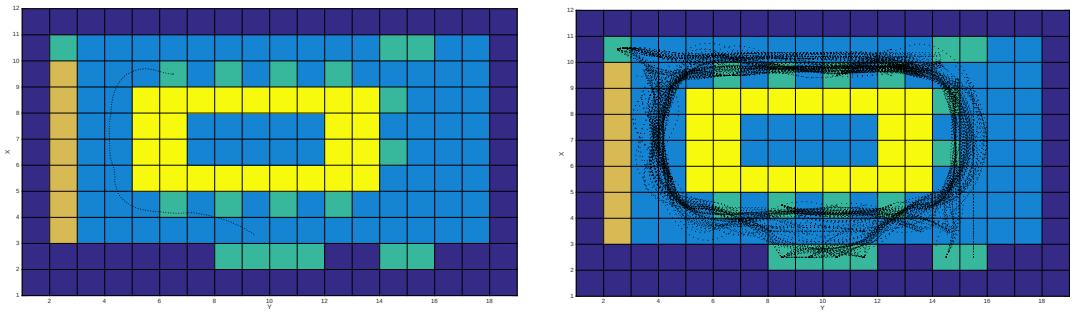
(A) (B)

FIGURE 5.2: Example environment. (A) Captured photo, (B) 2D mapping, where yellow: half height obstacles, green: entrance/exits or points of interest, light blue: traversable areas and orange & dark blue: full height obstacles/ non-traversable area.

Using a two dimensional cartesian representation of an environment we define a movement space in terms of x and y coordinates. The mapping represents a low resolution view of the environment, where one unit square maps to a set square measurement in the environment (e.g $0.5m$). Obstacles in the environment are also represented in the map, allowing the agents in the simulation an awareness of what is traversable and what

is not. Figure 5.2 (A) shows an example environment in which a meeting room is shown with a set of tables arranged in the middle of the room, Figure 5.2 (B) shows the 2D environment map of the same scene.

Using the defined entrance/exit locations and points of interest, an exhaustive set of paths that take into account all possible path connotations can be made. For each one of these paths a simulation is run in which an individual agent traverses the environment from a start location to a destination, avoiding any obstacles that may be in their way. Simulations are run according to the algorithm specified in Section 5.3.3.



(A) (B)

FIGURE 5.3: Simulated paths. (A) individual path, (B) overlay of all possible path connotations.

The results of an individual simulation provides positions $P = [x, y]^t \mathbb{R}^2$ for an agent at a specific time t. By removing the time component and plotting these points in the movement space we build up a picture of a traversed path (Figure 5.3 A), similarly plotting all the paths from all the agents demonstrates areas that are most commonly used (Figure 5.3 B). 2D histograms are created by binning each individual position, of each agent, at each time, into its respective unit measure within the environment map 5.2. Where $i \in [1 \dots X]$ where X is the maximum bin (unit measure) along the x axis of the environment map, and $j \in [1 \dots Y]$ where Y is the maximum bin (unit measure) along the y axis.

$$h_{i,j}(P) = \begin{cases} 1, & \text{iff } P_x \in i \text{ and } P_y \in j \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

The sum of points within each bin is used as a measure of frequency, 5.3. The result is a low resolution frequency map, indicating areas of high and low interaction (Figure 5.4).

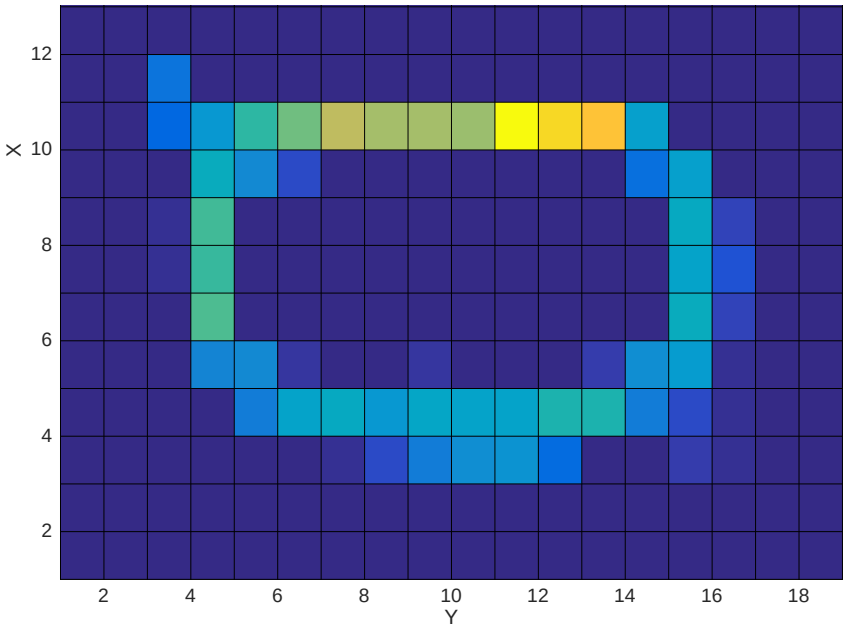


FIGURE 5.4: Resultant interaction map as a result of the simulated paths and the binning process.

$$\mathbf{H}_{i,j} = \sum_{k=1}^{t} h_{i,j}(P_k) \tag{5.3}$$

Thus locations of high frequency define areas in the environment in which the simulation algorithm estimates a higher level of human presence, as such these areas would present more of a risk than those of low frequency.

5.3.2.2 Visibility Maps

Using the same simulation techniques and histogram principle as 5.2 and 5.3, the concept is expanded to encompass the visibility component of simulated agents in an environment. Each agent within the simulation has a number of defined properties, these include agent radius, movement speed, acceleration and turning speed. In addition to these a number of properties are defined that pertain to that agent's ability to *see* the environment. A field of view is defined $\mathbf{F} = [\varphi, q] \in \mathbb{R}^2$ subject to ν, specifying the angular range of that agents peripheral vision φ, as well as a viewable radius q

$$\nu = f(\mathbf{F}_{\varphi,q}) \tag{5.4}$$

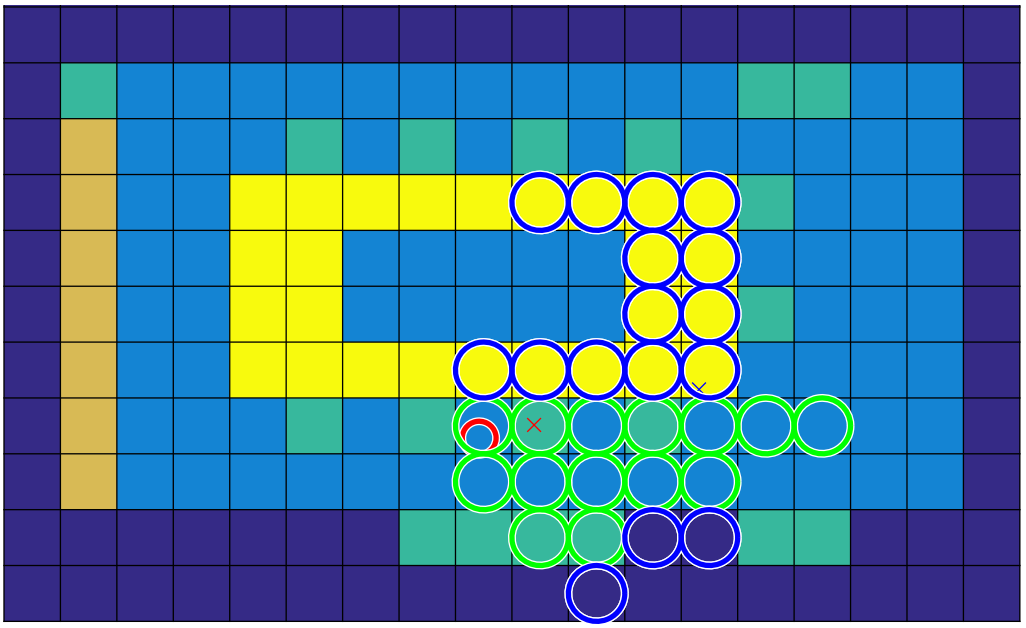


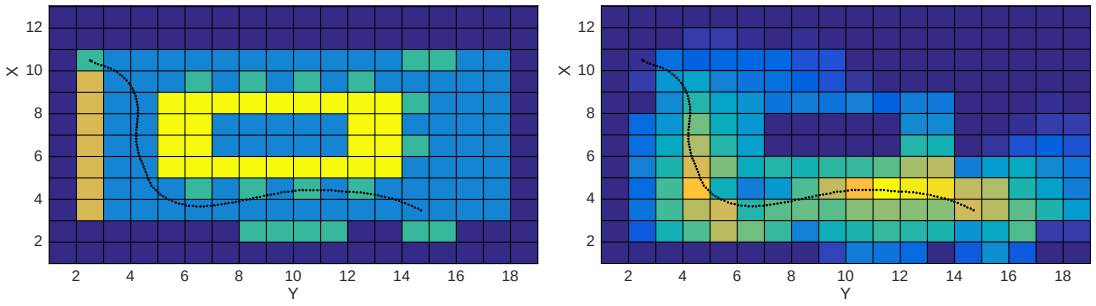
FIGURE 5.5: Example frame of a simulation with the agent's current field of view overlaid. Yellow circles indicate seen traversable area, blue indicates seen obstacles. The red circle is the current position of the agent and the cross is the direction of movement.

Where ν is a value from from zero to one defining how well an agent can see at a specific point of their vision, whereby closer and more central points are better seen and those at the edge of an agent's vision and further away rated worse. The visual field is represented by a binocular 180^o view. Visual acuity is based on a logarithmic scale [167], however a more comprehensive model of visual acuity in human peripheral vision could be applied.

During the simulation the agent's viewable area is recorded per time instance, subject to an agent's properties. Within the context of the environment mapping this is defined as whether an agent can see a specific unit square of the map or not (Figure 5.5).

The environment map is updated to reflect the differing heights of obstacles, such that a label is defined for obstacles that can be seen over and for obstacles that cannot (Figure 5.2 B). For example, walls fully obscure the agents view, however low height obstacles such as tables block vision directly behind them but allow vision further away. As can be seen in Figure 5.5, where the area in the centre of the tables cannot be seen as it is occluded by the presence of the tables, where as the table across is still visible to the agent.

As before simulations are run for the given connotations of paths. Position is extended such that $P = [x, y, \nu]^t \in \mathbb{R}^3$ where ν represents how well that position was *seen* by the agent at that time (subject to 5.4), as they navigate through the scene on their estimated



(A) (B)

FIGURE 5.6: Visibility Maps. (A) Individual path on which the visibility map is generated, (B) the visibility map produced from the agents field of view during the taken path.

path (Figure 5.5). Now instead of just the position of the agent in a scene contributing to the histogram, their entire field of view contributes to the histogram each frame. The contribution to each bin is made by the visibility component 5.5. In the case of Figure 5.6 C, this would mean 30 visible units of the map would contribute based on the how well it is viewed by the agent.

$$h_{i,j}(P) = \begin{cases} P_\nu, & \text{iff } P_x \in i \text{ and } P_y \in j \\ 0, & \text{otherwise} \end{cases} \tag{5.5}$$

The visibility map is then given as the summation of the histogram bins over time as 5.3.

Figure 5.6 (A) gives an example of a single agent's track with the compounded visibility map for that path shown in Figure 5.6 (B). The same method is used for all the given path combinations and a single compound visibility map returned for that environment (Figure 5.7).

The higher the visibility values recorded for an individual unit square of the environment map, the more often and better seen that area of the environment is likely to be. Conversely those areas with low values are not well observed and could further add to a hazard at that location due to it's potential to go unseen. As visibility is a positive measure and the risk scores to date are negative measures, the current histogram needs normalisation and inverting 5.6. As a result the histogram, $\mathbf{H}_{norm}$, then provides a measure of invisibility, within a range of zero to one.

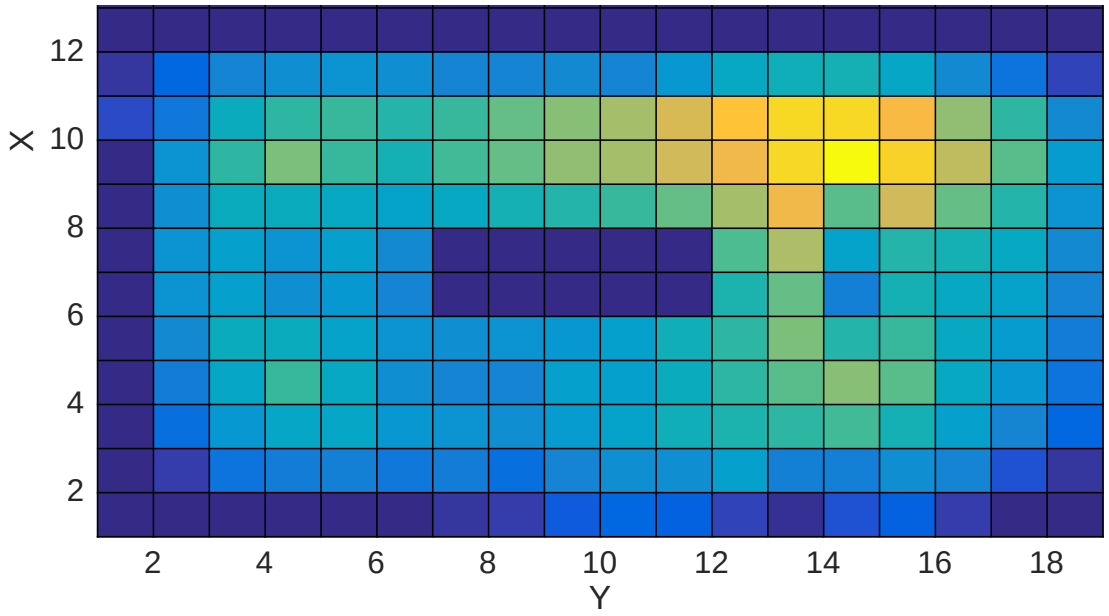


FIGURE 5.7: Resultant visibility map as a result of the simulated paths seen in Figure 5.3 (B).

$$\mathbf{H}_{norm} = 1 - \frac{\mathbf{H}}{\|\mathbf{H}\|} \tag{5.6}$$

Parameters for field of view and peripheral vision acuity allows the tailoring of the simulations to better reflect those that use the environment. As in the cases of children, the visually impaired, or the elderly who have reduced peripheral vision acuity or Tunnel Vision due to conditions such as glaucoma or brain damage amongst others.

5.3.2.3 Risk Avoidance Maps

In the same way that simulation can be used to predict the likely paths through an environment, it can also be utilized to predict the path likely to be taken when a risk is discovered. As with the visibility component, the same two dimensional environment map is used taking into account the height of obstacles within the scene. In addition to this, a hazard is defined at some location within the environment and is given a risk score. The use of a risk score rather than a binary, risk / no risk, classification allows the consideration of a human's ability to evaluate whether a risk poses a threat to them. This provides the mechanism by which an agent can decide to continue along their original path or take some avoiding action as a result of the discovered risk. For example, broken glass in a kitchen, an adult may well decide to continue their path

narrowly avoiding the glass, however in the presence of children they may wish to take a route that avoids the glass entirely.

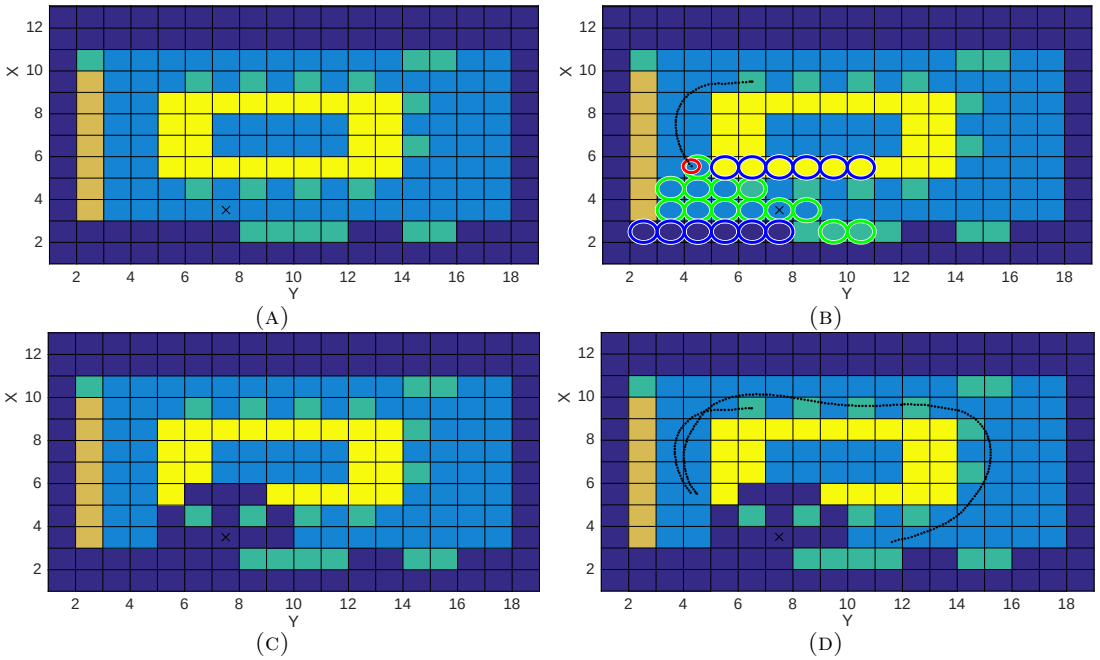


FIGURE 5.8: Risk avoidance. (A) Library map with black cross representing the risk in the scene, (B) the path the agent has taken until the agent sees the risk. (C) Updated environment map, creating an exclusion zone around the located risk. (D) The final taken path as a result of the rerouting process.

Detection of the risk is based on the field of view F of an agent during simulation. As in real world examples the user must be made aware of the risk before avoiding action is taken. If the simulated agent sees the risk during their navigation of the scene (Figure 5.8 B), and decides to reroute, then the simulation algorithm defines a obstacle/hazard area around the location of the risk (Figure 5.8 C) and computes a new path to follow (Figure 5.8 D). A detailed explanation of the decision making process is given is Section 5.3.3.

As a result of the rerouted path, the interaction and visibility maps for the given environment will be affected. Figure 5.9 shows interaction and visibility maps for the path demonstrated in Figure 5.8 against the same path without the presence of a risk. As can be seen, the areas where both interaction and visibility are highest, changes considerably based on the reroute.

The ability to compute these likely changes in environment interaction allows flexibility of the system, providing better contextual awareness of what might happen given a change in circumstance.

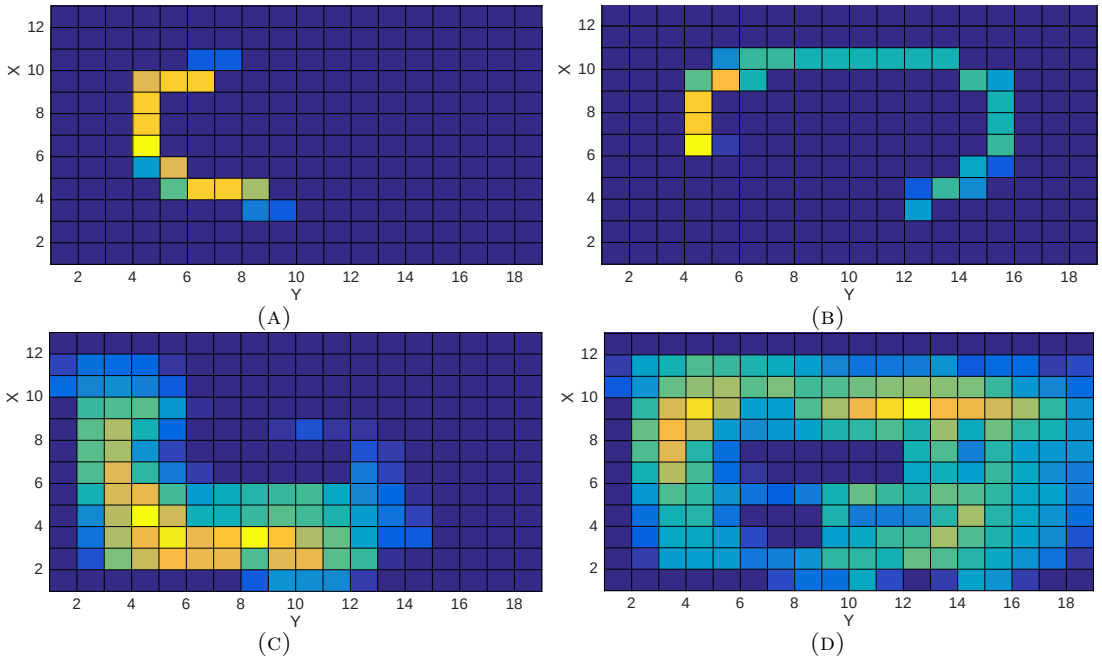


FIGURE 5.9: Changes to the visibility and interaction maps. (A) Interaction map for the direct path. (B) Interaction map for the rerouted path. (C) Visibility map for the direct path. (D) Visibility map for the rerouted path.

The final risk element E for the environmental risk maps is given as a combination of the risk associated with the presence of a person and their ability to see hazards in a given environment. As such a weighted combination of the histograms produced by the interaction maps H_I and the visibility maps H_V is given as

$$E = w_V \mathbf{H}_V + w_I \mathbf{H}_I \tag{5.7}$$

Where w represents the histograms contribution to the the final risk element and the two contributions $w_V + w_I = 1$. The weighting is flexible based on the final implementation allowing the condition of those using the environment to be considered during simulation.

5.3.3 Simulation

Within Sections 5.3.2 - 5.3.4, simulation algorithms are used to replicate human behaviour in an environment. The following section outlines the simulation algorithm utilised.

The model implementation used in the simulation of human behaviour in an environment is based on a combination of simulation algorithm models. Firstly a steering simulator based on the work of [10, 100] in which the concepts of simple crowd behaviours such as separation, object avoidance and agent collision detection are utilised. These have been implemented with the social forces model structure in which each of these elements produce a force applied to the agent to adjust their movement vector. The magnitude of these forces is based on distance. An additional step, using a planning simulation methodology, based on the work of Karamouzas et al [103] is used as a predictive collision detection algorithm to produce natural agent avoidance within the simulations, this again is implemented by the application of a force upon the simulated agent. As outlined in 5.8.

$$F_a = g_a - p_a + \sum_{i=1}^{n} f(a, b_i) + \sum_{j=1}^{m} f(o_j) + \sum_{k=1}^{o} f(a, b_k) \tag{5.8}$$

where g_a is the current destination along the path of the agent a to its final goal, with p_a being the agent's current position. The forces for separation, $f(a, b_i)$, object avoidance $f(o_j)$ and the predicative agent avoidance $f(a, b_k)$, is calculated for any relevant entity within a defined neighbourhood.

In Section 5.3.2 an environment map is created for the scene. When the simulation begins, an agent performs a route plan using the A* algorithm to estimate the most direct course from their start location to their destination. A* is used as it provides a near optimum path through the environment whilst keeping computation cost down [168]. Given that the intended applications for this work are likely indoor environments and may well be computed within the confines of a domestic robot, considerations as to the speed of runtime and algorithm applicability are important. Additionally A* is computationally efficient enough that it can be run on demand in real time if required.

To allow an agent to navigate a scene with unknown risks, a decision making process is required. In any given situation, the action an agent will take is defined by a set of probabilities. In this situation the agent decides to either continue on their existing path or recalculate a new one to avoid a risk. As path finding and an agent's movement are

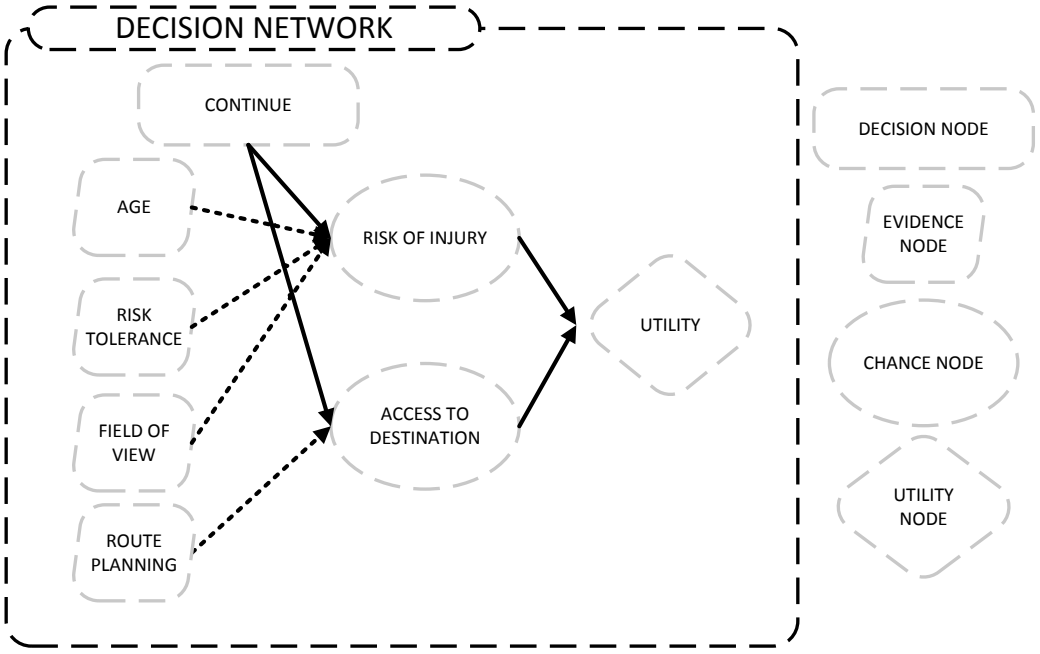


FIGURE 5.10: Decision network for risk interaction

governed by the parameters defined in 5.8, it is only this risk related choice that requires a decision making process.

Let **S** be a state at a given time in a simulation for an agent. Figure 5.10 outlines this decision process for state **S** as a decision network. Within the diagram the decision node represents the problem to be resolved by a set of actions A. In this case which action to take whilst traversing the environment (continue or reroute). The chance nodes are indicative of the probabilistic outcomes associated with this decision represented as probability distribution functions. In this case the chance of injury and the chance of being able to find a path to the destination. The evidence nodes represent the knowledge the agent has, which directly affects the chance nodes. For example the risk of injury will change based on an agent seeing a hazard. Finally the utility node is the utility or value for the state **S** based on the chance nodes and an agent's preference.

To determine the best action for the agent to take, the principle of maximum expected utility (MEU) is utilised 5.9. Here the possible actions A are assessed using the expected utility EU, based on the known evidence **E**. Therefore the MEU represents the action A_i which is classified as most favorable by the agent.

$$MEU(A_i|\mathbf{E}) = \max \arg_A \sum_{i=1}^{a} EU(A_i|\mathbf{E}) \tag{5.9}$$

Evidence in this case is a combination of agent variables and environmental feedback such that $\mathbf{E} = [p_a, p_r, p_p, p_f]$ where p_a is the age of the agent, p_r is their tolerance of risk (represented by gaussian distributions, assuming prior knowledge of the means and standard deviations of the relevant information), p_p is the presence of an alternative route and p_f represents a seen hazard. In this simulation environment there are two actions that can be taken by an agent representing a single decision; continue on their preplanned path, or reroute to avoid the hazard.

The expected utility of an action is defined in 5.10. Expected utility represents a measure of both the likelihood of a particular state occurring, combined with the agent's preference for that outcome. Here a possible action A has a number of possible outcome states $\text{Result}_i(A)$. For each outcome a probability is assigned based on the evidence $\mathbf{E}$. $\text{Do}(A)$ represents the supposition that the action A is executed in the current state.

$$EU(A|\mathbf{E}) = \mathbf{P}(\text{Result}_i(A)|\mathbf{E}, \text{Do}(A))U(\text{Result}_i(A)) \tag{5.10}$$

The given probability of each action A is then multiplied by a utility function U for the possible outcome states. In this example the utility function is simplistic as only a limited number of states exist based on the decision network presented in Figure 5.10. The utility associated with accessing the destination will nearly always take precedence; if the direct route to the destination is accessible then the agent will continue, only in cases where risk of injury is high will the agent decide to reroute. However if the route is blocked then regardless of the risk of injury the agent will have to reroute.

As an example, in a normal situation with the absence of risk, the probability associated with risk of injury is low and the probability for reaching the destination is high. Given that the agent wants to get to the destination, whilst avoiding injury, the $EU(A|\mathbf{E})$ for the A to carry on the current path is high. However if the evidence changes and a risk is detected through an agent's field of view, the probability of risk of injury increases.

FIGURE 5.11: Frames of source CCTV footage and generated video using the composition techniques.

If there exists an additional route the agent can take to avoid the risk, the $EU(A|\mathbf{E})$ for the A to reroute will be higher and therefore the preferred option.

Utilising the social force model and the higher level MEU based decision model a detailed simulation of the human behaviour corresponding to navigating an environment and dealing with risk is created.

5.3.4 Simulation Evaluation using Compositing Techniques

The following section will explain in detail the various aspects of the proposed Human and Group Behaviour Simulation Evaluation framework. The framework provides a method of simulation algorithm evaluation that rates how realistic the human walking behaviours look compared to sample footage. Evaluation can be done on a frame by frame basis or on a sequence as a whole, providing flexibility in how the simulation is evaluated. Additionally the proposed methodology requires no track or path information for the source material, allowing any pedestrian video footage captured from a static viewpoint to be used as source material. Evaluation of an algorithm's performance is key to defining how realistic the simulation outputs are. In the case of the interactions maps, how effective a simulation is at replicating human behaviour directly impacts the accuracy of the produced risk evaluation.

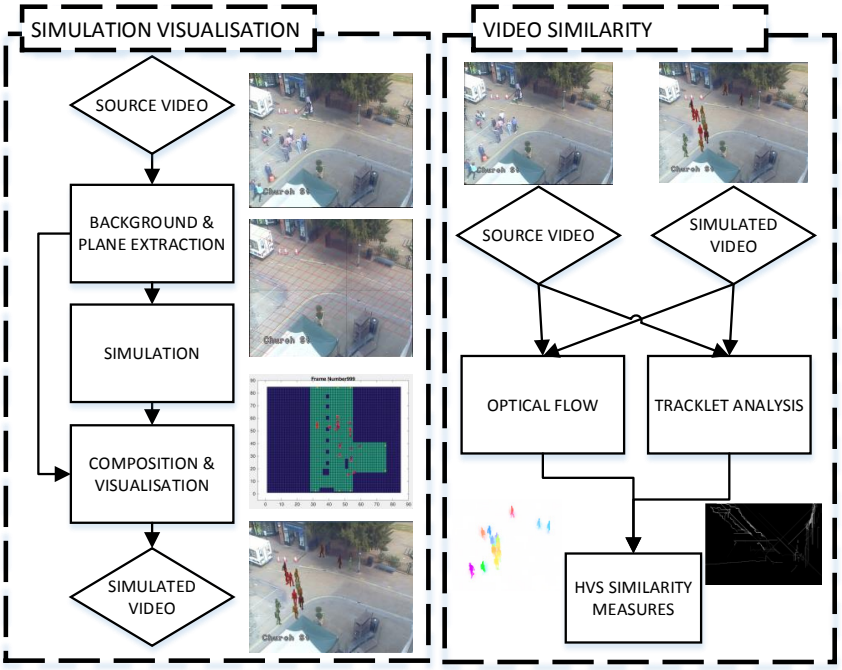


FIGURE 5.12: Overview of the Human and Group Behaviour Simulation Evaluation framework.

Comparison is made using the original source footage and a video created using composition techniques, in which simulated agents are superimposed into the background of the source video data. A set of metrics designed to evaluate the visual similarity of the two videos is used to provide a quantifiable similarity metric. These are designed to emulate the way the Human Visual System (HVS) perceives motion, both in direction and volume. Additionally the principles of Weber's Law [160] are included to better match the metrics to the way humans see. Weber's Law states that the eye senses illumination, within a range, logarithmically. As such, we combine this principle with well known computer vision based video analytics techniques to create HVS features which allow for comparison between video footage.

Fundamentally the framework is made up of two components; simulation visualisation and video similarity. Both aspects are designed to be modular, supporting the inputs of any simulation algorithm and any video analytic features. However the proposed HVS features provide a good generalisation of simulation evaluation requirements in a broad range of situations. Figure 5.12 provides a brief overview of the process. Further detail of each section will follow.

As the framework compares video data to derive a similarity value, firstly a simulated

video must be constructed. Using the source video sequence, the background is extracted. Additionally a two dimensional plane is extracted representing a top down view of the given scene. Simulations are run to produce paths for virtual agents to follow based on the extracted plane. The visualisation component is used to composite the extracted 2D background image and 3D rendered agents as they follow the simulated paths. Frames are output from the visualisation into a final simulated video sequence (Figure 5.11). Once both a simulated and source video are available, the similarity can be evaluated. Optical flow and tracklet analysis are run and features extracted from the subsequent data. Finally a distance measure is used to analyse the difference in features to give the final similarity metric.

5.3.4.1 Background and Plane Extraction

To allow the composition of the simulated video to be created, the background of the source video sequence is required. To obtain this let I be an individual frame of the source video, and B the background of that source video, defined as the mean value of each pixel throughout the whole sequence.

$$B(x, y) = \frac{\sum_{i=1}^{n} I(x, y, i)}{n} \tag{5.11}$$

Where x, y are the pixel location and i the current frame of the source sequence, with n being the total number of frames in the sequence. Other methods based on Gaussian mixture models could be used in order to obtain more accurate results [169].

Once the background image has been subtracted the process of defining the perspective grid is applied. The perspective grid allows scale mapping of an environment from the viewpoint of the source video camera pose. The resultant grid represents a top down environment map of the viewable area. This mapping is used during simulation and can be utilised for the work in Section 5.3.2. Using the concept of perspective scale along a line we can, through the definition of two parallel lines that run to the vanishing point of an image, estimate distance in arbitrary units of measure within this perspective space (Figure 5.13 B). This unit can be based upon an object in the scene with known dimensions or using pedestrians [170].

Initially the user defines the points **i** and **j**, in the 2D image space, forming a line along an edge that leads to the vanishing point of the image. A second line is defined by the points **k** and **l**, such that it runs 'parallel', relative to the vanishing point in the 3D space of the captured image, to the line defined by points **i** and **j** (Figure 5.13 A).

At a location along the line **ij** the user defines another point $\mathbf{u}_1$, such that the line $\mathbf{iu}_1$ represents the unit of distance m from which all further perspective points are defined. An additional point $\mathbf{u}_2$ is defined along the line **ik** which represents the same relative distance in 3D space as m.

For the next step of the proposed algorithm the reference points $\mathbf{T}_{vanish}$, $\mathbf{R}$, $\mathbf{R}_0$ and $\mathbf{T}_{n-1}$ are initialised automatically (Figure 5.13 A).

In more detail, the vanishing point $\mathbf{T}_{vanish}$ is defined as the point at which the lines **ij** and **kl** intersect, this may well be at a position outside of the image plane. As such $\mathbf{T}_{vanish}$ is defined as

$$\mathbf{T}_{vanish} = f(\mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{l}) \tag{5.12}$$

An arbitrary point $\mathbf{R}$ is selected at a random location outside the triangle $\mathbf{iT}_{vanish}\mathbf{k}$. The point $\mathbf{T}_{n-1}$ is defined as the point of intersection of the lines $\mathbf{iR}$ and $\mathbf{kT}_{vanish}$

$$\mathbf{T}_{n-1} = f(\mathbf{i}, \mathbf{R}, \mathbf{k}, \mathbf{T}_{vanish}) \tag{5.13}$$

Finally the point $\mathbf{R}_0$ is defined.

$$\mathbf{R}_0 = f(\mathbf{u}_1, \mathbf{T}_{n-1}, \mathbf{R}, \mathbf{T}_{vanish}) \tag{5.14}$$

With these points initialised a recursive algorithm is applied to calculate equidistant points along the line $\mathbf{iT}_{vanish}$ in 3D space. As the user has already defined the first of these points $\mathbf{u}_1$, for the purposes of the recursive step, these will be relabeled as $\mathbf{G}_n$.

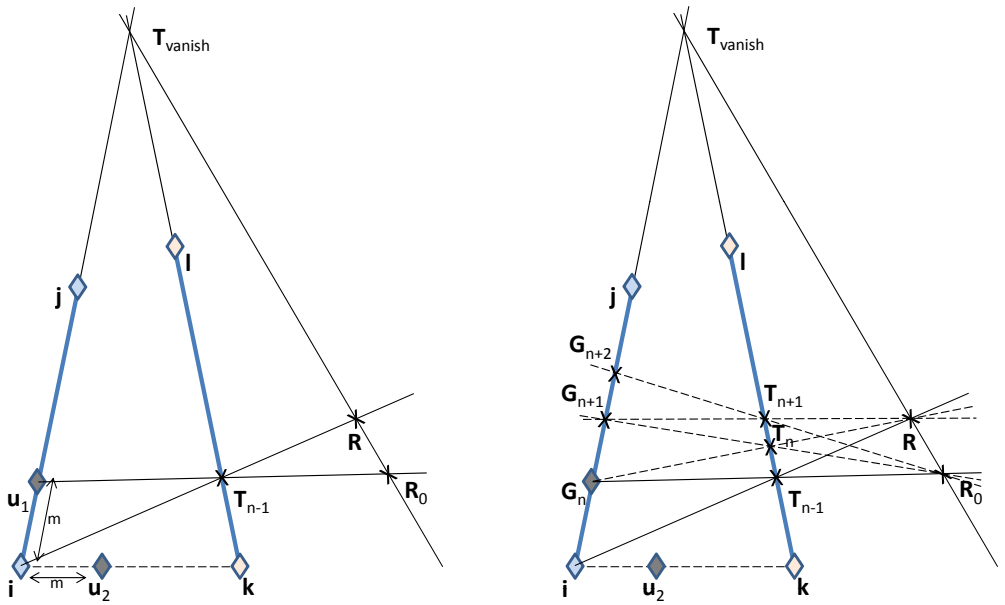


FIGURE 5.13: (A) User defined points and initialisation. (B) The first two iterations of the recursive algorithm.

This is a two-step iterative process, with the point $\mathbf{T}_n$ being defined as the intersection of the lines $\mathbf{G}_n\mathbf{R}$ and $\mathbf{kT}_{vanish}$.

$$\mathbf{T}_n = f(\mathbf{G}_n, \mathbf{R}, \mathbf{k}, \mathbf{T}_{vanish}) \tag{5.15}$$

and during the second step the next equidistant point $\mathbf{G}_{n+1}$ on the line $\mathbf{iT}_{vanish}$ is expressed as a function of

$$\mathbf{G}_{n+1} = f(\mathbf{R}_0, \mathbf{T}_n, \mathbf{i}, \mathbf{T}_{vanish}) \tag{5.16}$$

This process is repeated until $\mathbf{G}_{n+1}$ is no longer within the borders of the original background image.

The grid is initially defined using all the equidistant points on the line **ik** using the distance $\mathbf{iu}_2$ as a unit. Lines are defined between each of these points and the vanishing point $\mathbf{T}_{vanish}$ of the image. The scale points $\mathbf{G}$ are plotted along each of these newly defined lines forming the grid. Additionally if required, the recursive process can be inverted to create points moving away from the vanishing point. This ensures that the

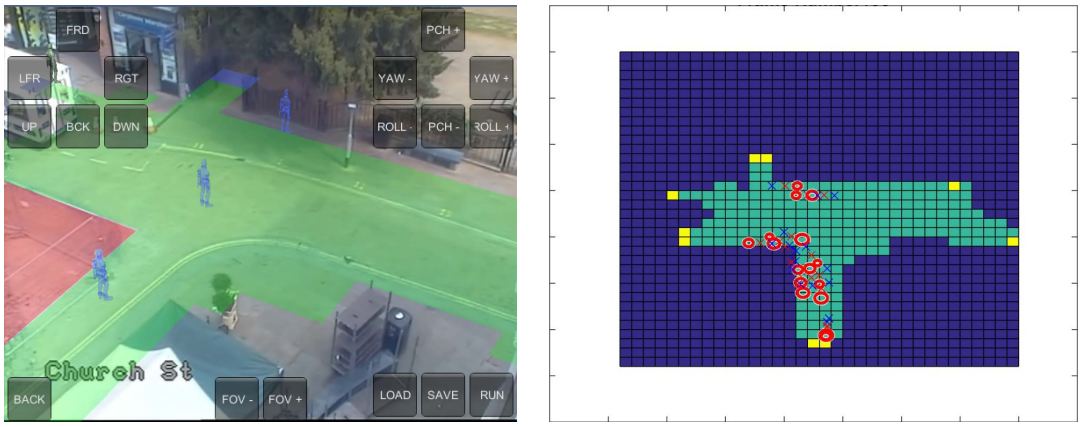
FIGURE 5.14: Resultant perspective grid overlayed on the original source image.

entire image plane is encapsulated by the defined grid, regardless of where the user has defined their points.

The resultant grid represents the perspective plane of the source image. On that grid the areas (cells) with obstacles (i.e. cells where pedestrians cannot walk) are annotated as is information about entrance/exit locations. In order to help the user; the obtained grid is superimposed on the extracted background image (Figure 5.14). Here red cells indicate areas where agents can walk, white represent obstacles and green marks an entrance or exit. This annotated version of the perspective plane is then used as the ground plane by the simulation algorithms.

5.3.5 Composition and Visualisation

The visualisation stage of the framework performs the composition of a scene utilising the extracted background obtained from the source video and the generated perspective plane (Section 5.3.4.1). The key to a visually similar composition is the positioning of a virtual camera at the same location as in the original scene. By using layers the camera can have the source image as a background and the visualised 3D agents controlled by the simulation superimposed. Due to this, it is important to line up the perspective



(A) (B)

FIGURE 5.15: (A)Example composition of the Kvan scene with test agents and perspective floor plan. (B) Example visualisation of the simulation running for the Kvan scene.

plane with the background to give the illusion of the agents walking through the scene. This alignment can be performed manually using the position and orientation of the camera or automatically using camera calibration techniques [171, 172].

Sample agents are then placed in the scene at various locations to ensure that the perspective and scaling parameters of the agents are appropriate to the scene. Figure 5.15 (A) demonstrates this with agents in blue positioned at different locations in the scene. These values can be adjusted manually or calculated automatically using the methods provided in [173]. In Figure 5.15 it can also be seen that the imported floor plan is coloured according to each cell's defined values, green meaning areas the agents can walk, blue defines entrance exit points and red would represent obstructions that will obscure the agents when they are located behind. To control the agents in the scene, position and orientation information is required for each frame. This is obtained using the desired simulation algorithm and the same perspective grid map (Section 5.3.3).

As the goal is to create videos with similar crowds, a number of parameters from the original video are required. Using the source video sequence an analysis is made of the pedestrians in the scene, outlining paths and estimated crowd density. For this work the information was extracted manually or provided by the datasets used, however work exists to help automate this process [128, 130]. Once all the required parameters for the agents are defined the simulation is run and outputs recorded at the frame rate of the original source video (Figure 5.15 B).

Finally, with the composition completed and the simulations run the visualisation of the scene is performed. Agent models are created and sized according to the obtained

parameters. For each frame of the simulation the agent location and rotation is updated based on the simulation algorithm output and a composite frame is captured. Once visualisation is completed the individual frames are compiled into a video sequence. Importantly the resolution and the number of frames in the new composite video should be equal to that of the source video.

5.3.6 Simulation Similarity Metrics

Once visualisation of the composite video is complete, the source and the simulated video sequences are used to extract features in order to measure their level of similarity. These features are based on the optical flow and tracklets of the moving objects in both sequences.

5.3.6.1 Optical Flow and Tracklet Estimation

An optical flow method tries to calculate the motion between two image frames at times t and $t + \delta t$ at each pixel position [121]. Let a pixel at location (x, y, t) with intensity $I(x, y, t)$ be moved by δx, δy and δt between the two frames, the following image constraint equation can be derived.

$$I(x, y, t) = I(x + ax, y + ay, t + at) \tag{5.17}$$

Assuming the movement to be small enough, the image constraint at $I(x, y, t)$ can be developed using the Taylor series in each dimension.

$$I(x + ax, y + ay, t + at) = I(x, y, t) + \frac{\delta I}{\delta x}\delta x + \frac{\delta I}{\delta y}\delta y + \frac{\delta I}{\delta t}\delta t \tag{5.18}$$

Thus the equation below is derived where Vx, Vy, are the x and y components of the velocity or optical flow of $I(x, y, t)$ and Ix, Iy and It are the derivatives of the image at (x, y, t) in the corresponding directions.

$$I_x V_x + I_y V_y = -I_t \tag{5.19}$$

The solution as given by Lucas and Kanade is a non-iterative method, which assumes a locally constant flow. Assuming that the flow (Vx, Vy) is constant in a small window, of size m x m with $m > 1$, centred at pixel x, y and numbering the pixels as $1...n$, a set of equations is obtained.

$$\begin{bmatrix} I_{x1} & I_{y1} \\ \vdots & \vdots \\ I_{xn} & I_{yn} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} -I_{t1} \\ \vdots \\ -I_{tn} \end{bmatrix} \Rightarrow A\vec{M} = -b \Rightarrow \vec{M} = (A^T A)^{-1} A^T (-b) \tag{5.20}$$

This means that the optical flow can be found by calculating the derivatives of the image in all three dimensions. A weighting function $w(i, j)$, with $i, j \in [1, \ldots, m]$ is added to give more prominence to the centre pixel of the window. Gaussian functions are preferred for this purpose, but other functions or weighting schemes are also possible. For computing local translations, the flow model can be extended to affine image deformations. Black and Anandan in [122], describe how the single motion assumption, as well as the constant brightness constraint are not always valid. They discuss how these assumptions can be relaxed in order to develop a more robust estimation framework.

Tracklet estimation is a well researched topic with many algorithms available in the literature. These can be based on motion or other features and utilise particle and Kalman filters [115, 116, 174, 175]. Specifically, the problem of motion based tracking can be split into detecting moving objects in each frame and the association of those moving elements to a continuous corresponding object over time. Gaussian mixture models are used to apply background subtraction and the noise is eliminated using morphological operations on the obtained foreground mask.

In the case of Kalman filters, the track's location in each frame is predicted and a likelihood of a detection is assigned to each track. The Kalman filter is a recursive estimator, meaning that only the estimated state from the previous time step and the current measurement are needed for computation of the current state. The Kalman filter has two distinctive features; firstly its mathematical model is described in terms

of state-space concepts; Secondly, the solution is computed recursively. Usually, the Kalman filter is described by a system state model and a measurement model.

The Kalman filter works by evaluating measurements over time, in this case to predict the track of a pedestrian in a crowd from frame to frame. Using a statistical model of the properties of that pedestrian and the previous state a prediction is made for the location of the pedestrian in the next time frame. The next measurement is made and a comparison between the predicted and actual states is made. Using this analysis the model is updated, to improve prediction going forward. Weighting is applied to this update process which favours estimates with higher levels of accuracy relative to the current measurement. The process is then repeated using the updated model to predict the position of the pedestrian in the next time frame.

Optical flow and tracklet estimation is an important aspect of this framework. In this system the optical flow algorithm proposed in [122] and the tracking method presented in [174] were utilized but the system is designed in such a way that allows the incorporation of multiple motion estimation or tracking methods as plugins. Based on this system architecture the proposed evaluation framework is dynamic and capable of utilizing current and future state of the art tracking methods.

5.3.6.2 Motion and Tracklet Flux Based Similarity Metrics

In order to evaluate the similarity level of the simulated and source videos a new metric is required that will allow an objective comparison incorporating the Human Visual System (HVS) based similarity features and metrics. Weber's Law [160] and the work in [161, 162] states that a human's ability to define motion as the point when the signal-to-noise ratio is regarded as at a stimulus intensity. Therefore, the minimum motion contrast dV as a function of background motion V, required for the human visual system to notice a change is expressed as:

$$dm = L\frac{dV}{V} \tag{5.21}$$

where dm is the differential change in motion perception, dV is the differential increase in the velocity and V is the velocity. The parameter L is to be estimated using experimental data. The proposed measure includes Fechner's Law [176], which relates velocity V, to perceived motion, $\mathbf{M}$, as seen by the human visual system, as follows:

$$\mathbf{M} = Lln(\frac{V}{V_{max}}) \tag{5.22}$$

where V_{max} is the 'upper threshold' of the human eye. The proposed metric is based on the motion and tracklet flux histograms obtained from the perceived motion $\mathbf{M}$ utilizing standard computer vision algorithms.

Let us assume that $I_R(\vec{u}, t)$ and $I_S(\vec{u}, t)$ are the image frames of a real and the correspondent simulated scene, respectively. The motion vectors for each pixel location in each frame are estimated using the optical flow techniques shown in 5.23 and 5.24

$$M_R(\vec{u}, t) = f(I_R(\vec{u}, t), I_R(\vec{u}, t-1)) \tag{5.23}$$

$$M_S(\vec{u}, t) = f(I_S(\vec{u}, t), I_S(\vec{u}, t-1)) \tag{5.24}$$

The estimated tracklets are obtained using motion information and Kalman filters.

$$T_R(n_R, \vec{u}, t) = f(M_R, I_R) \tag{5.25}$$

$$T_S(n_S, \vec{u}, t) = f(M_S, I_S) \tag{5.26}$$

Since the motion vectors and the tracklets are available the histogram of oriented optical flow (HOOF) [177] is calculated both for the real and simulated scenes.

$$f_R^{HOOF} = HOOF(M_R) \tag{5.27}$$

$$f_S^{HOOF} = HOOF(M_S) \tag{5.28}$$

Also, a 2D histogram of the motion parameters is obtained using 5.29 and 5.30.

$$f_R^{H2D}(r_{ij}) = m_{ij}(M_R) \tag{5.29}$$

$$f_S^{H2D}(r_{ij}) = m_{ij}(M_S) \tag{5.30}$$

Where r_{ij} is the i^{th} and j^{th} motion level in an interval, and m_{ij} is the number of pixels in all the given frames whose motion level is r_{ij}. Regarding the tracklets, the time parameter in 5.25 and 5.26 is removed by superimposing all of them at the same time instance. The similarity metric here can be applied on any given time interval, which can be the whole sequence or a small time fragment. In the same way as in 5.29 and 5.30 we obtain:

$$f_R^{T}(r_{ij}) = m_{ij}(T_R) \tag{5.31}$$

$$f_S^{T}(r_{ij}) = m_{ij}(T_S) \tag{5.32}$$

Finally, the flux of the features in 5.27 - 5.32 is represented by the surface integral of the given vector field.

$$\Phi(\vec{u}, t) = \Sigma_{\vec{u}} \Sigma_t f d\vec{u}\ dt \tag{5.33}$$

Based on 5.33, we obtain Φ_R^{HOOF}, Φ_S^{HOOF}, Φ_R^{H2D}, Φ_S^{H2D}, Φ_R^{T} and Φ_S^{T} that correspond to the proposed HVS features. All the features can be applied either on the whole sequence or on smaller blocks allowing specio-temporal adaptation of the proposed features and metrics. In order to measure the similarity and rank the algorithms, a distance measure is utilized e.g. Correlation, Bhattacharyya, Chi Square, Histogram Intersection, Dot Product, L1, Euclidean or earth mover's distance (EMD). For this work Bhattacharyya is utilised due to its use in similar work [131].

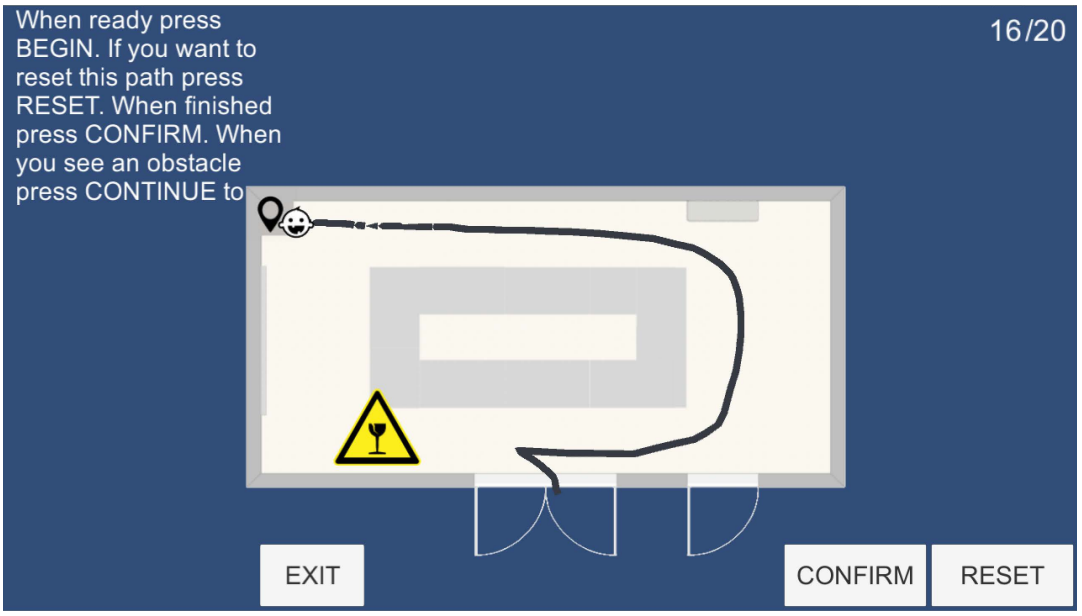


FIGURE 5.16: Tablet application with example scene in which the participant discovers a risk in their original path and is forced to reroute.

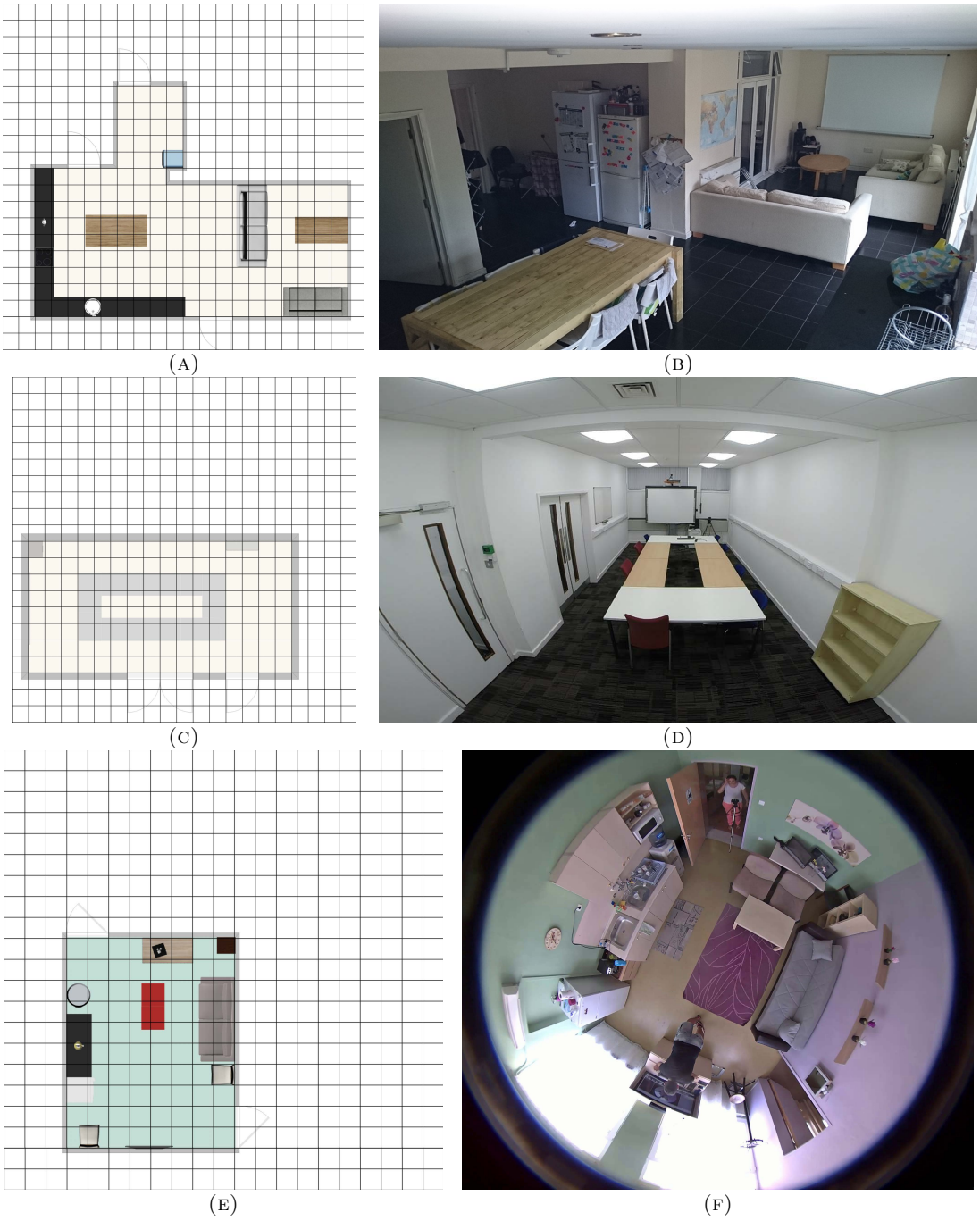
5.4 Results

5.4.1 Experiment Environment

5.4.1.1 Environmental Risk Maps

To evaluate the accuracy of the predicted areas of high interaction, tests must be conducted to ensure the simulation algorithm produces results that are in line with human paths. To achieve this, ten participants were asked to navigate virtual environments extracted from real world locations (Figure 5.17 A & B) using a tablet application. Within the application the user is asked to drag a representation of a person around a top down floor plan (Figure 5.16). The recorded paths represent the ground truth from which future simulations can be tested against. Each participant was asked to navigate 20 paths in two different environments. Fifteen of the paths were a direct start location to goal example, with participants asked to avoid any obstacles that might stand in their way. The remaining five examples had the addition of a risk located within the scene which only appeared to the user when they were in close enough proximity. On discovery they were asked to reroute within the environment to find a new path to their destination.

Further comparison is made using long term observation data found here [14]. Data is collected over 12 months in which the tracks of people interacting with a lounge room is



(A) (B) (C) (D) (E) (F)

FIGURE 5.17: Floor plans and image of the three environments used in the simulations. (A-B) Kitchen. (C-D) Library. (E-F) Lounge.

recorded continuously throughout the day (Figure 5.17 E-F). Using the track information the same interaction map can be produced. Individual tracks from one interest point of the room to another is not available, instead the dataset holds position information for a user at time frames throughout the 12 month period. Due to this data format a like for like simulation is not possible. However the dataset provides areas of most interaction through the use of clustering techniques.

5.4.1.2 Human and Group Behaviour Evaluation Framework

To evaluate the proposed Human and Group Behaviour Evaluation framework, a total of five different scenes were used from various crowd datasets (Mall Dataset [134], PETS2009 [13] and RBK [136]) and captured crowd and pedestrian videos sequences. Scenes of different environments including both indoor and outdoor spaces, with a large range of camera orientations and crowd configurations. Additionally the frame rates of the videos varied from less than 10fps up to 24fps providing a challenging and diverse set of scenes from which to evaluate both the simulation algorithm and the effectiveness of the evaluation framework.

5.4.2 Environmental Risk Map Evaluation

To evaluate the validity of the environmental risk map methodology, 200 simulations were conducted per environment for the kitchen and library rooms (outlined in Section 5.4.1.1) using the same start positions and destinations as specified for the human participants. Additionally simulations were run for the lounge room using the clustered areas of interest and entrance/exit points from the dataset. In this case a total of 330 simulations were run to try to gain better coverage of the same long term observations in the dataset. The simulation algorithm was responsible for navigating the environment, avoiding obstacles and any discovered risk.

Using the resultant interaction maps a comparison is made between those generated by the human participants and those by the simulation algorithm. Using the cosine distance, a measurement is made comparing the similarity of the two histograms, created using the techniques outlined in Section 5.3.2. Here a low distance represents interaction maps that are similar and which represent realistic generation of paths. Comparisons are broken down into direct routes and routes in which a risk was discovered. Agent values defined for the simulation are based on existing work used by Asano et al. [95, 111] and changed periodically to fit the scenario as required.

Table 5.1 demonstrates the measured distance between the generated interaction maps. The maps are produced from the real data (captured using the tablet application and as a result of the long term study of elderly adults) and the data produced as a result of the simulation algorithm. Each histogram was tested against the others produced, allowing

TABLE 5.1: Measured distance between produced interaction maps (no risk rerouting), using the Cosine similarity.

Data Source	Library Sim	Kitchen Sim	Lounge Sim
Library Real	**0.385**	0.580	0.815
Kitchen Real	0.663	**0.327**	0.658
Lounge Real	0.873	0.698	**0.498**

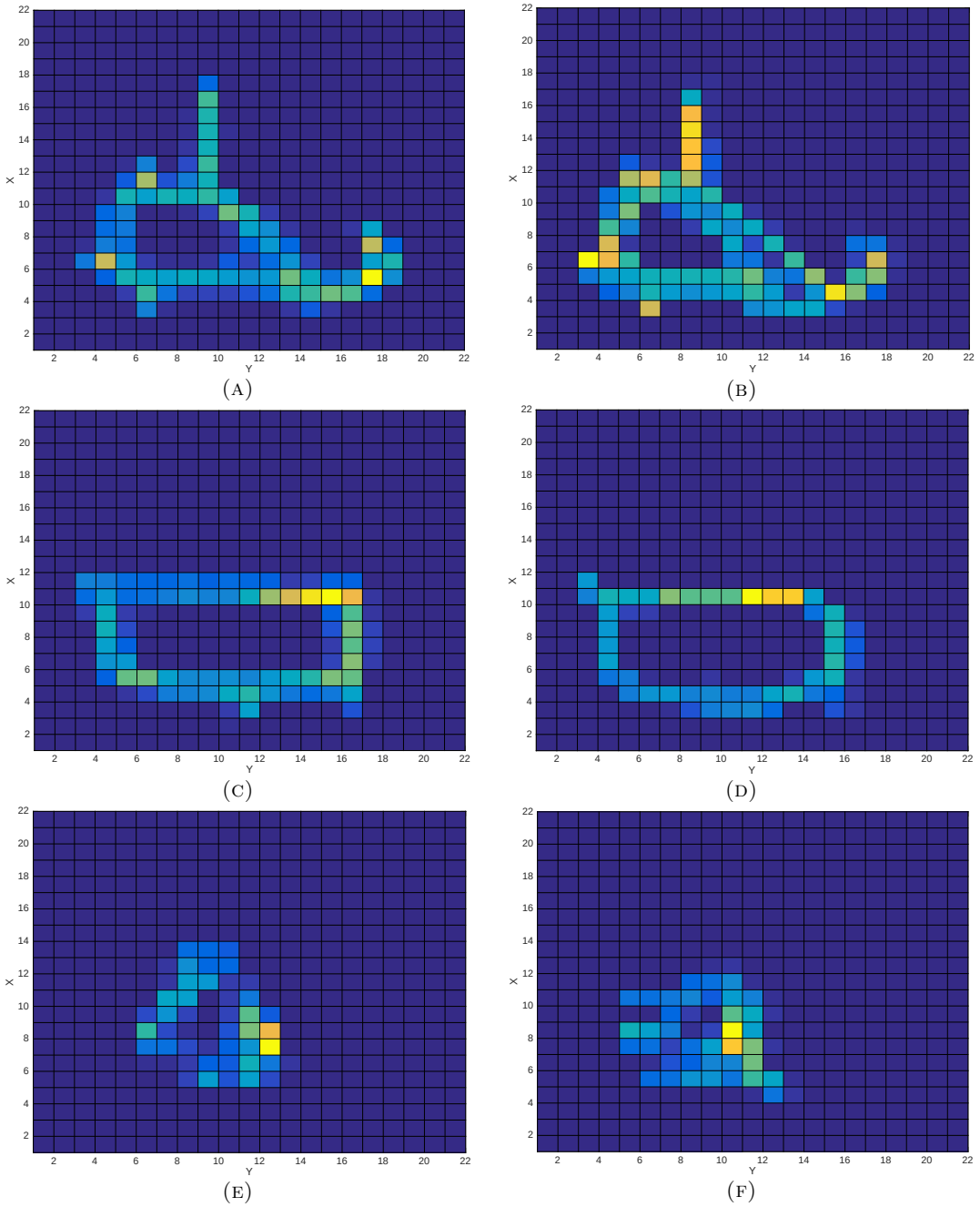


FIGURE 5.18: Interaction maps for the three scenes. (A-B) Real and simulated maps for the Kitchen. (C-D) Real and simulated maps for the Library. (E-F) Real and simulated maps for the Lounge.

TABLE 5.2: Measured distance between produced risk avoidance maps, using the Cosine similarity.

Data Source	Library Sim	Kitchen Sim
Library Real	**0.369**	0.522
Kitchen Real	0.591	**0.329**

measurements between different environments to be tested. Logically the simulation data should be closest to the real data of the same environment, which is demonstrated in the results. Figure 5.18 shows the visualised interaction maps for each environment. Each unit square corresponds to $0.5m^2$ across all images, this represents a reasonable level of detail to represent the room, and for comparison the images were orientated so as to be as visually similar as possible. The kitchen and library environments are visually more similar to that of the lounge and this is reflected in the results. In the conducted simulations for the lounge data, a total of 330 individual agent paths were simulated for the environment.

The same result structure is presented for the risk avoidance maps (Table 5.2). In this case data from the lounge dataset was not available for reroutes due to a risk. As such, the data captured using the tablet application and the ten participants is compared against simulated tracks produced using the simulation algorithm outlined in Section 5.3.3. A strong similarity is seen between real and simulated data for the same environment (Figure 5.19) with a noticeable difference being measured between data from different environments. From these results it can be seen that the simulation algorithm produces similar tracks and decision making ability as that of the human participants. Through this validation it can be seen that with the simulation algorithm's ability to accurately replicate human behaviour the resultant visibility maps generated provide a realistic view of the scenes risk.

Using the estimated agent paths, visibility maps can be generated to produce a final risk map for an environment based on the simulated visibility of agents as they navigate the environment. Figure 5.20 show the visualised compounded vision maps generated as a results of the simulation algorithm. Based on these, any defined hazards within the scene can be updated with this additional risk information.

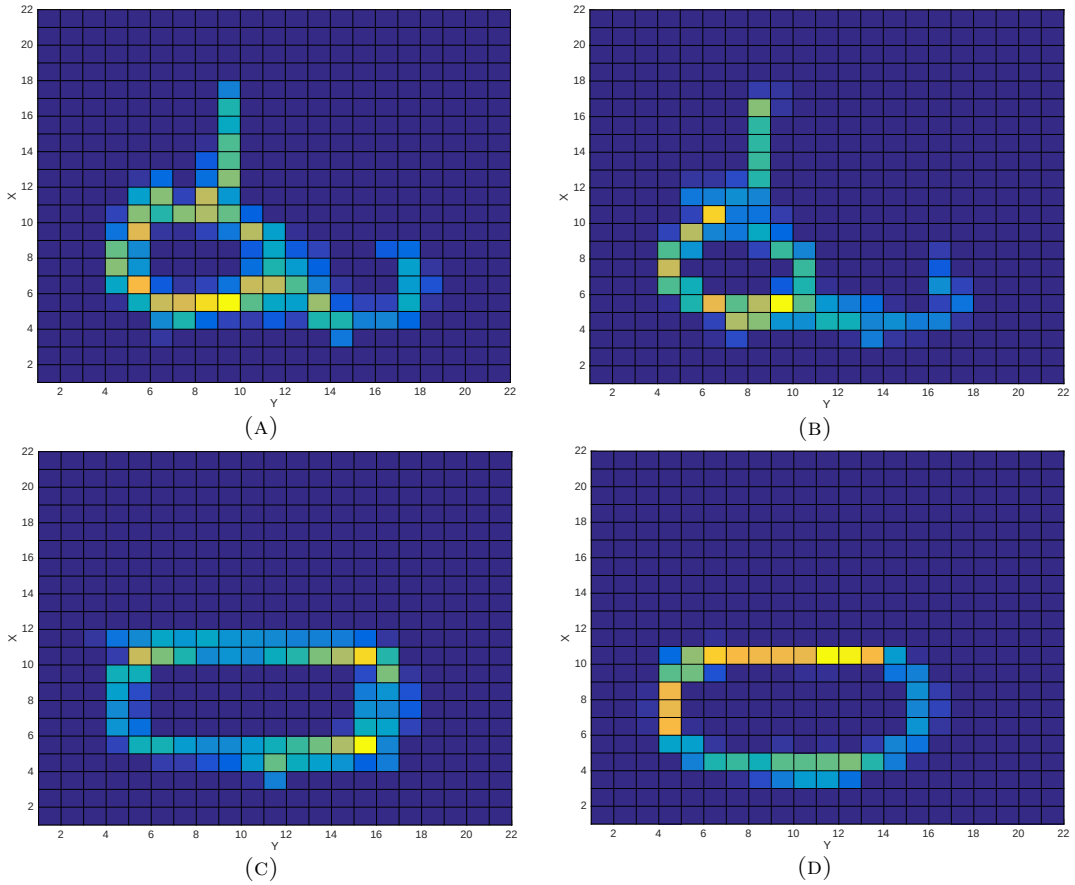


FIGURE 5.19: Risk avoidance maps. (A-B) Kitchen real and simulated. (C-D) Library real and simulated.

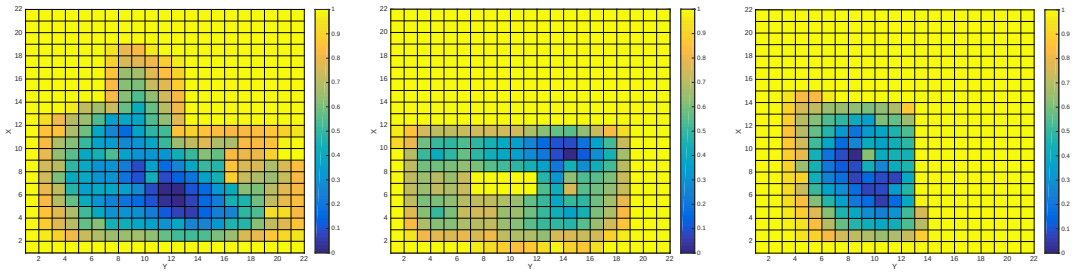


FIGURE 5.20: Visibility maps based on the agents field of view for the three scenes. Areas of lighter yellow represent areas of low visibility and therefore higher risk.

5.4.3 Simulation Evaluation

To evaluate the Human and Group Behaviour Simulation framework composite videos for five scenes were created. For each scene videos were created using four different levels of agent speed and three different population levels, totalling 12 simulations and resultant composite videos per scene. This provides an accurate assessment of the proposed framework and the relative features. Figure 5.21 presents example source and simulated frames for a few scenes.

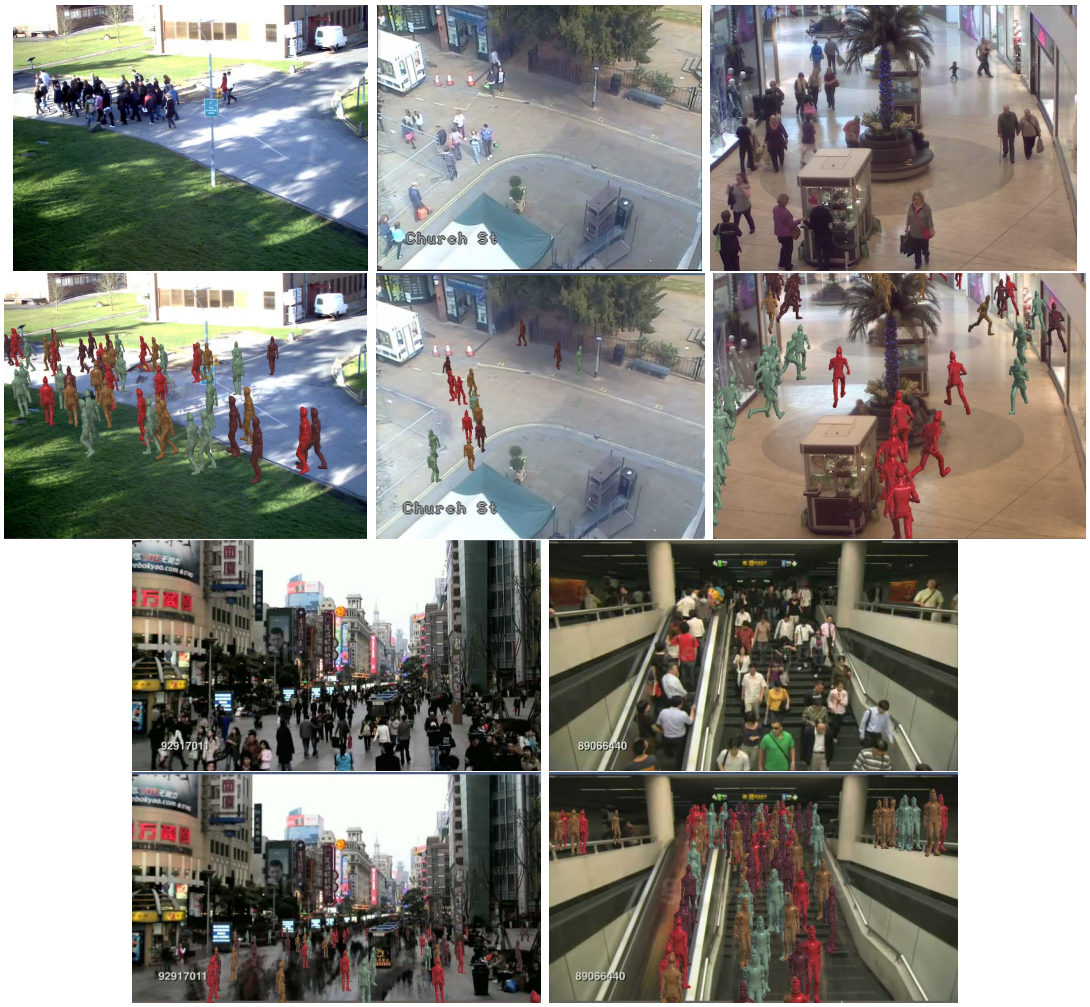


FIGURE 5.21: Example source and simulated frames. Row 1 (Source): Road, Kvan. Row 2 (Simulated): Road, Kvan. Row 3 (Source): Mall, Krad1, Krad2. Row 3 (Simulated): Mall, Krad1, Krad2.

For each scene the background image was extracted and the perspective grid defined. Simulations were performed for each of the cases mentioned above and the outputs used to create composite visualisations for each. Simulated videos were then created with the same frame rate, length and resolution as the source videos. The simulation themselves are run at a set frame rate (50 frames per second), a desired frame rate is also specified allowing for the movement of the agents to be visualised at the same frame rate as the source video, without having to adjust the agent properties between simulations.

Each simulated video, and its respective source video, had the optical flow and tracklets estimated using the methodologies outlined in Section 5.3.6.1. Finally the distance measures were used to compare each visualisation against its source. Three measures were used in the comparison, the tracklets (Φ_R^T and Φ_S^T), Histogram of Orientated Optical Flow per frame(Φ_R^{HOOF}, Φ_S^{HOOF}) and the Histogram of Orientated Optical Flow for the

TABLE 5.3: Average Bhattacharyya distance between source and each simulated video sequence, across all five Scenes for Φ^{HOOF} features.

	Agents Speed			
# Agents	Very Slow	Slow	Same	Fast
Few	10.43	7.93	9.19	7.83
Same	10.34	9.27	7.23	7.54
Many	10.85	10.10	10.28	10.51

TABLE 5.4: Average Bhattacharyya distance between source and each simulated video sequence, across all five Scenes for Φ^{H2D} features.

	Agents Speed			
# Agents	Very Slow	Slow	Same	Fast
Few	3.26	2.78	2.68	3.07
Same	3.50	3.21	3.12	3.41
Many	3.84	3.66	3.64	3.90

TABLE 5.5: Average Bhattacharyya distance between source and each simulated video sequence, across all five Scenes for Φ^{T} features.

	Agents Speed			
# Agents	Very Slow	Slow	Same	Fast
Few	5.73	5.82	4.67	4.78
Same	6.81	3.46	2.44	5.00
Many	7.86	5.87	6.38	5.92

TABLE 5.6: Average Bhattacharyya distance between source and each simulated video sequence, across all five scenes for the feature combination.

	Agents Speed			
# Agents	Very Slow	Slow	Same	Fast
Few	6.35	5.39	4.39	5.10
Same	6.76	5.26	4.14	5.19
Many	7.30	6.42	6.79	6.65

sequence(Φ_R^{H2D}, Φ_S^{H2D}). The HOOF features and 2D Histograms used a window size of 64×64 pixels per frame. Using these features, a generalised statistical measure of the differences in movement from the source human behaviour to the simulated agents is defined. The distance metric used to compare the features is the Bhattacharyya Distance [178]. For these experiments no pre-defined groundtruth is required, instead each scene has its number of agents and their speed estimated. It is expected that simulations that have a similar number of agents and relative speed to the source video will produce the lowest distance measure.

Tables 5.3 - 5.6 contain the average distance measures, after applying the equation 5.33, across all five scenes for 12 different simulations. As expected the lowest distance values are seen when the simulation parameters closely match those of the source material.

TABLE 5.7: Mean Opinion Score (MOS) of human observations of similarity.

	Speed of Agents			
# Agents	Very Slow	Slow	Same	Fast
Few	1.02	2.95	3.35	2.32
Same	1.92	3.27	4.36	2.87
Many	1.53	3.18	3.45	2.72

TABLE 5.8: Correlation (Pearson) between combination features distance and MOS, with and without Weber's Law applied.

	Metrics Without Weber			Metrics With Weber		
Metric	Avg	# Agents	Speed	Avg	# Agents	Speed
Φ^{HOOF}	0.67	0.49	0.70	0.54	0.31	0.62
Φ^{T}	0.59	0.46	0.60	0.63	0.59	0.57
Φ^{H2D}	0.24	0.06	0.41	0.28	0.02	0.44
Combined	0.55	0.36	0.60	0.61	0.44	0.65

To further evaluate the methodology, a group of ten people were asked to rate all simulated visualisations against their respective source material, with focus on evaluating the speed, number and track of the agents. Using the Mean Opinion Score (MOS) method, the participants were asked to provide a rating of one to five where five represented a high similarity to the source material and one a strong dissimilarity. The values for all the participants were averaged to give a score for each scene. This evaluation technique demonstrates how similar the proposed metrics and methodology reacts compared with the Human Visual System. The results are contained in Table 5.7.

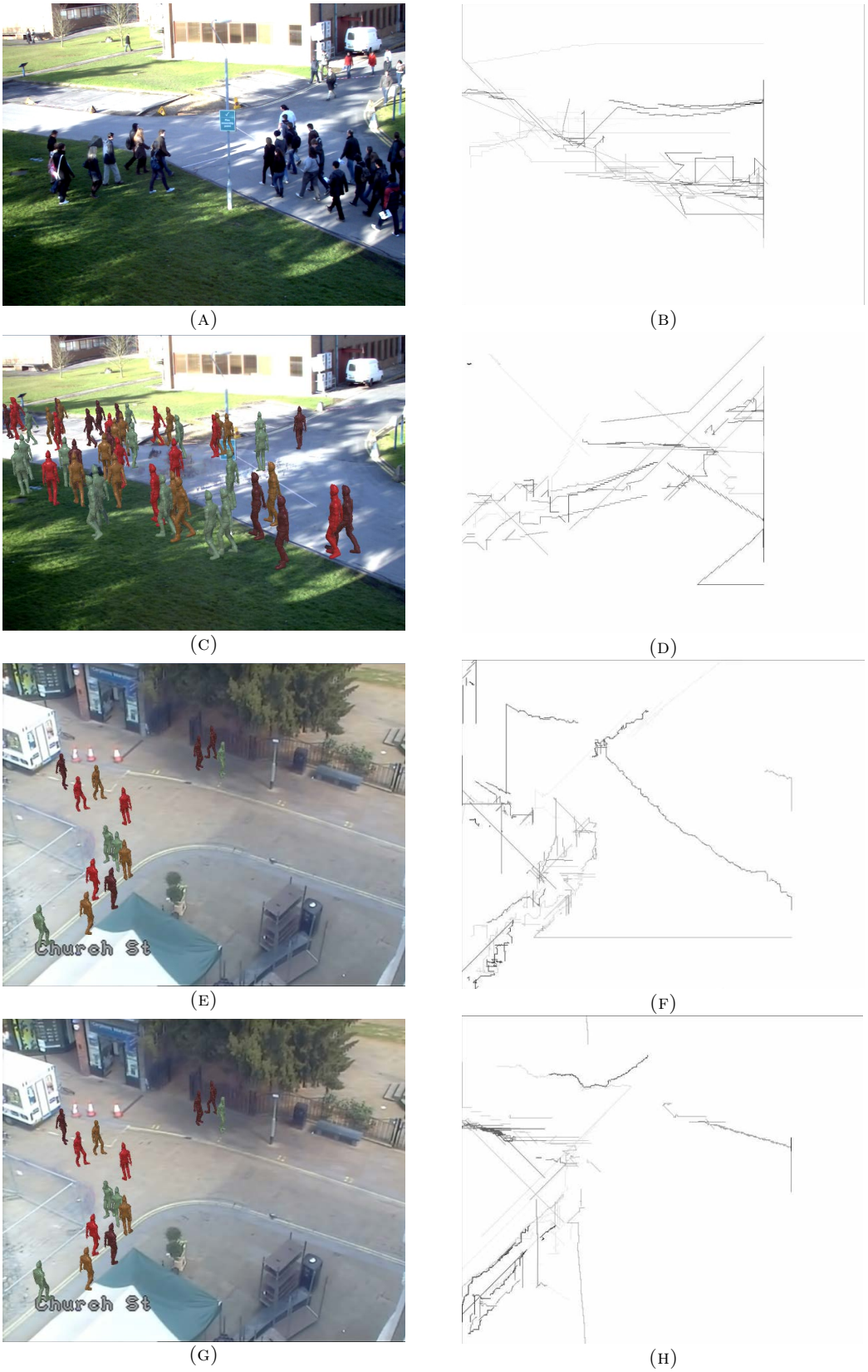
Table 5.8 is a breakdown of individual features against the human participant's ability to evaluate video properties. It can be seen that in certain instances the correlation between human and specific feature types is reasonably high. However by using the weighted sum of all three proposed metrics, and again comparing to the MOS, a more robust methodology is seen. This is not surprising as it is often observed that humans have difficulty distinguishing the difference between large amounts of slow moving agents versus a smaller amount of agents moving faster. As a result the combination of distance metrics from all three features more closely matches the Human Visual System's ability to evaluate motion. The weighting of the combination in this case is equal, however the optimal combination will be application dependant. Some metrics will perform better on different types of scenario. For example videos recorded from a lower point of view may not return descriptive tracklet information. Also within Table 5.8 we see the effect that incorporating Weber's Law into the features has. In all cases the combination of metrics better correlates to the human perception of movement in the videos.

By using the average distance from all of the three proposed features a robust system is demonstrated. However each of the individual feature provides a unique insight into the simulation accuracy. For example, evaluation of the tracklets allows an insight into how accurately the simulation model replicates the movements of the source material. The simulation model is outlined is Section 5.3.3. As such in complex scenarios where the source agents change direction a number of times, a strong dissimilarity is expected, likewise in more simplistic scenes where the simulation agents closely follow the source tracks a low dissimilarity is expected. A visual example is given in Figure 5.22, where we can see in the first scene (A-D) that there is an obvious visual difference between the source and the simulation, whereas in the second scene (E-G) the similarity is much higher. This is visualised using the tracklet plots which represent a compound image of the tracklets over the duration of the video.

Utilising the HOOF feature per frame and per sequence, an analysis of the amount of movement and magnitude of the optical flow can be made. Visualised examples of these two features are presented in Figures 5.23 & 5.24. Figure 5.23 is the compounded HOOF features for an entire sequence. The Figure 5.23 (A-B) represents the source material, with (C-D) being the simulation with similar values for number of agents and their speed. Figure 5.23 (E-F), (G-H) represent low and high levels of movement respectively. Figure 5.24 is the HOOF features using an individual frame. As before (A-B) is the source with (C-D), (E-F) and (G-H) being simulations with the previously mentioned parameters. In both cases its clear to see how the adjustment of speed and number of agents affects the output. Additionally effects on the tracklets can be seen. In the examples where the agent's speed is very low, parts of the scene are left unchanged by agent movement.

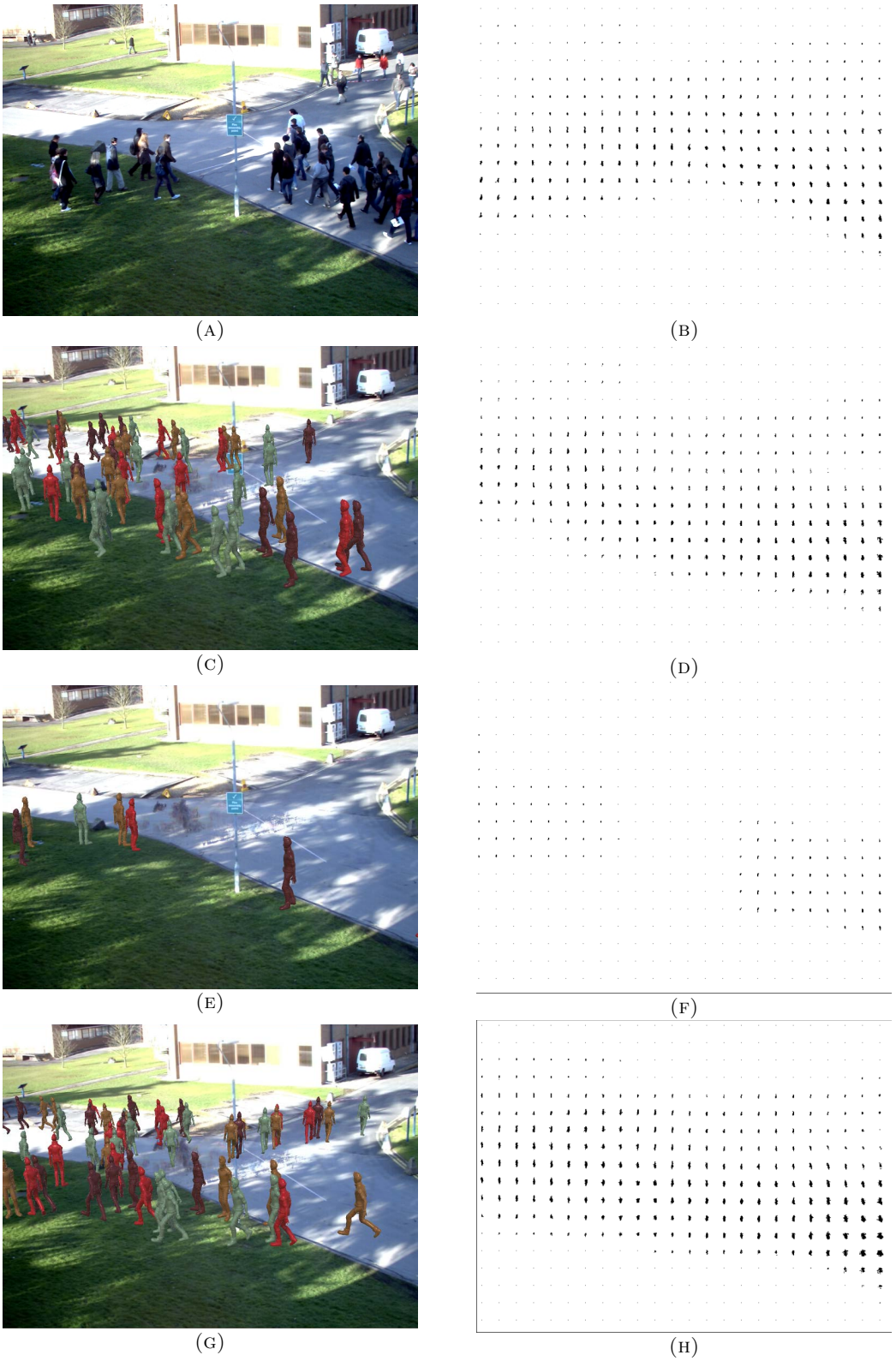
5.4.4 Risk Score

Finally to create the value for the risk element E, for use in the Risk Estimation framework the environment risk map is utilised, based on the interaction and visibility maps. Figure 5.25 show the visualised risk score for the given environments based on 5.7. Here the weightings $W_I = 0.75$ and $W_V = 0.25$ were selected experimentally, giving added credence to the interaction of the agent over the visibility. As both the visibility maps



(A) (B) (C) (D) (E) (F) (G) (H)

FIGURE 5.22: Side by side tracklet comparison for Road and Kvan. (A-B) Still from source Road video and tracklet. (C-D) Still from simulation and tracklet. (E-F) Still from source Kvan video and tracklet. (G-H) Still from simulation and tracklet.



(A) (B) (C) (D) (E) (F) (G) (H)

FIGURE 5.23: Histogram of Orientated Optical Flow per sequence using Road (Left example still from the video sequence and right, HOOF visualisation). (A-B) Source image, (C-D) medium, (E-F) low and (G-H) high speed and number of agent examples.

and the interaction maps are normalised the resultant environment map has risk scores for each unit area of between zero and one.

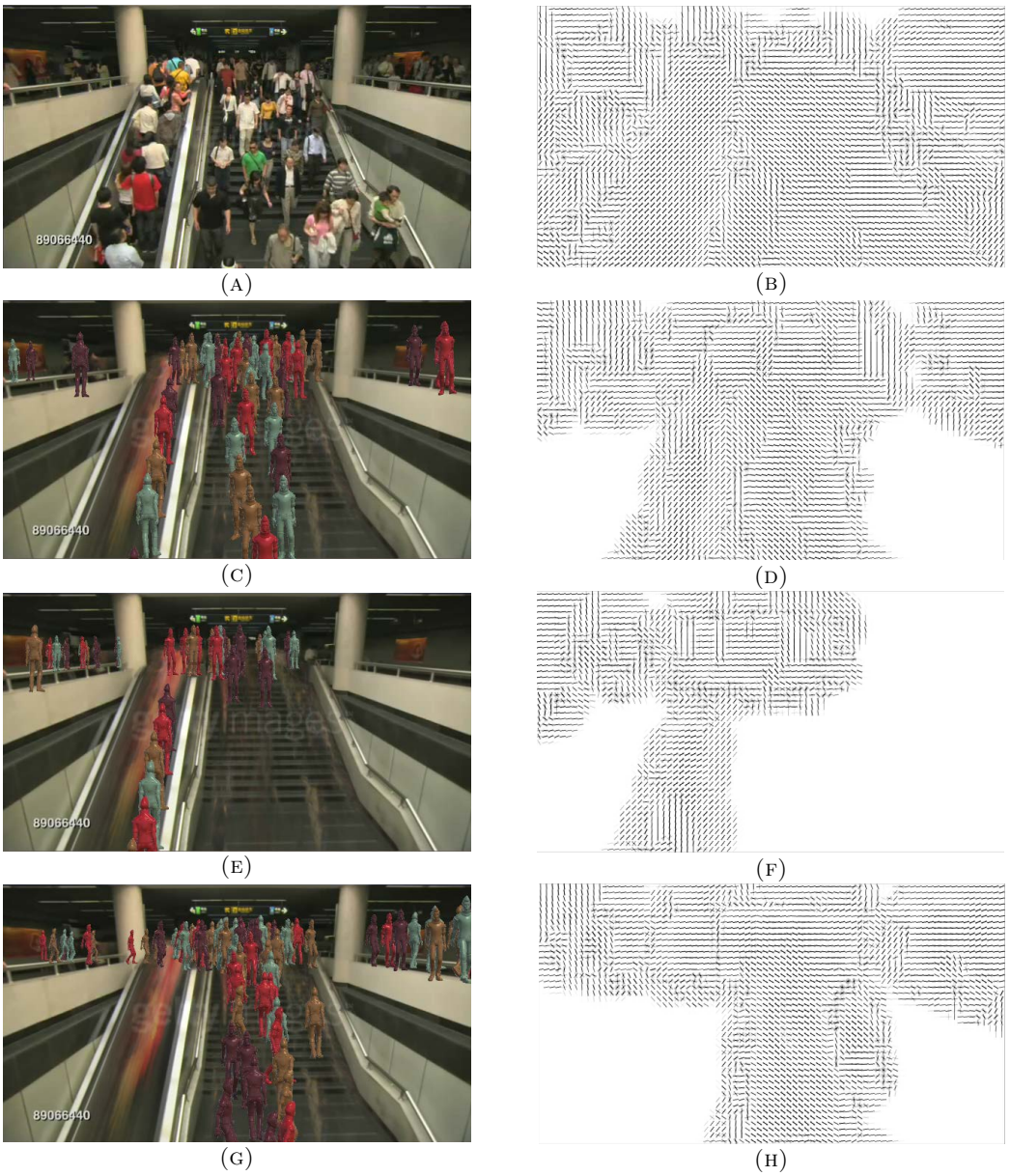


FIGURE 5.24: Histogram of Orientated Optical Flow per frame using Krad2 (Left: example still from the video sequence. Right: HOOF visualisation).(A-B) Source image, (C-D) medium, (E-F) low and (G-H) high speed and number of agent examples.

Using 5.1 we can get a final risk score comprising the stability, hazard features and environmental risk. Due to the nature of the environmental risk maps, the risk element contribution is for a specific area of a scene. To illustrate this using the results from one of the risk scenes evaluated in previous chapters (Figure 4.17) is placed into each of the three environments and the risk score updated. For each environment the same objects are placed in three different areas to demonstrate the effect the environmental aspects of risk have on the already calculated hazard and stability estimation results (Figure 5.25 left column).

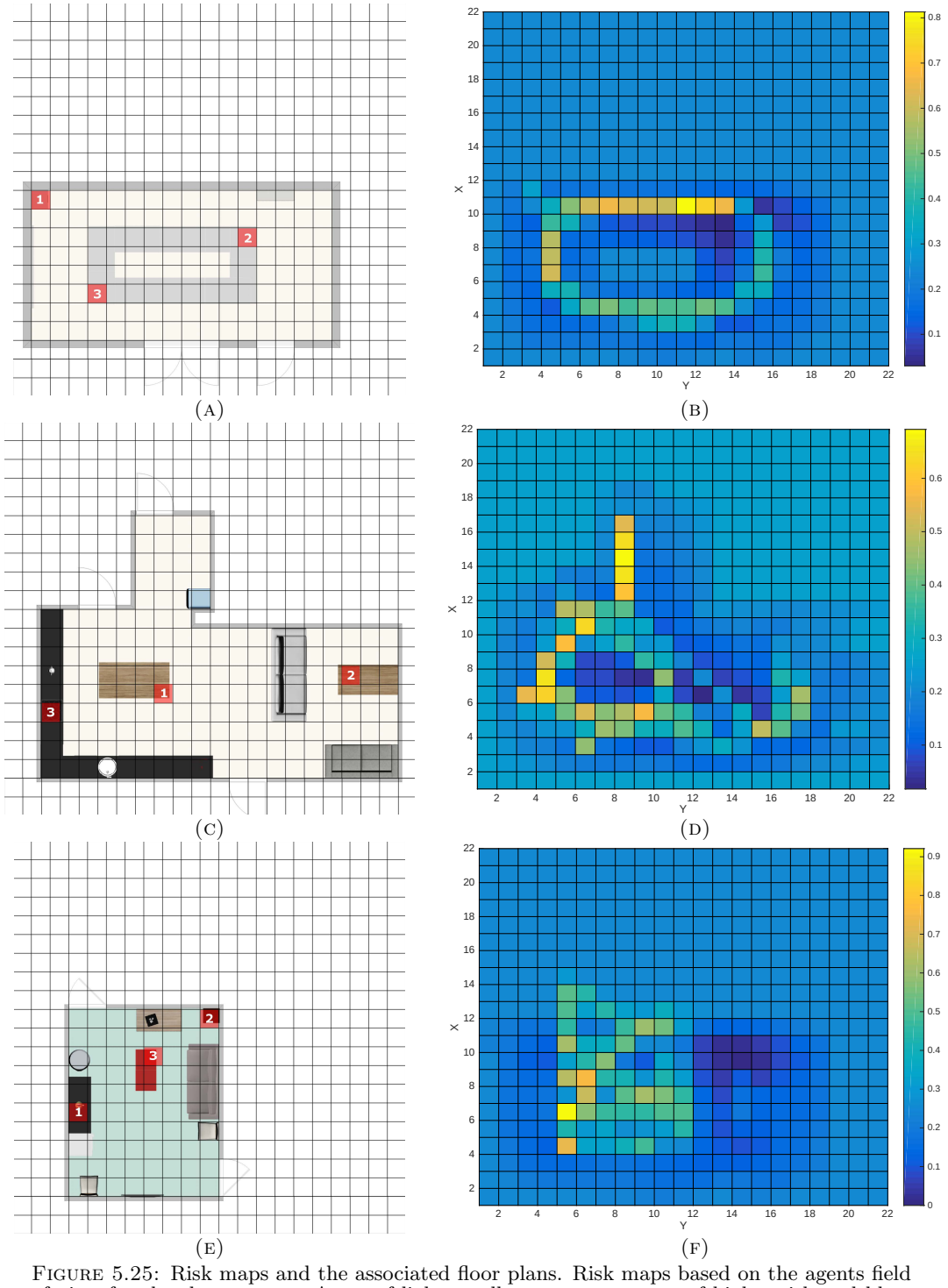


FIGURE 5.25: Risk maps and the associated floor plans. Risk maps based on the agents field of view for the three scenes. Areas of lighter yellow represent areas of higher risk and blue, areas of low risk. The locations of the placed risk objects in each scene is highlighted. (A-B) Library. (C-D) Kitchen. (E-F) Lounge.

TABLE 5.9: Risk scores based on 5.7, extracted from the environmental risk maps (Figure 5.25) for the three locations in each room.

	Location		
Room	1	2	3
Library	0.053	0.199	0.379
Kitchen	0.032	0.171	0.552
Lounge	0.175	0.301	0.529

TABLE 5.10: Final risk score considering stability, hazard features and the environmental risk maps.

	Location		
Library	1	2	3
Lv1	0.2940	0.3424	0.4025
Lv2	0.2904	0.3388	0.3988
Lv3	0.2555	0.3039	0.3639
Kitchen	1	2	3
Lv1	0.2870	0.3332	0.4601
Lv2	0.2834	0.3296	0.4565
Lv3	0.2485	0.2947	0.4216
Lounge	1	2	3
Lv1	0.3346	0.3766	0.4524
Lv2	0.3309	0.3729	0.4488
Lv3	0.2960	0.3380	0.4139

Table 5.9 shows the environment map risk score for the three highlighted points in each room (Figure 5.25 left column). The example locations have been selected to illustrate the range of risk presented in each environment. Location one is a low risk example from each scene with location three illustrating a higher risk. The highest risk locations tend to be where the simulation has concluded people walk the most. As the stability and hazard feature examples used here is based on a table, the selected locations represent tables in each respective room. Table 5.10 represents the final risk scores for the example risk scene in each of the presented rooms. Utilising the average hazard feature scores for the objects in the risk scene, results are broken down by a per stability level and per location risk score. In this example the weightings for each of the risk elements is sent to 0.3333, thus representing an equal contribution from each element.

5.4.5 Conclusion

To further extend the risk elements applicable to the Risk Estimation framework, interaction maps were developed to evaluate the effect that humans have on the risks in an environment. Simulation techniques were utilised to build environmental risk maps which highlight the areas most visible and most commonly visited in a specific previously unknown environment. Tests were conducted to evaluate the similarity of the paths generated by the simulation algorithm presented in Section 5.3.3 to those created by human participants. Additionally comparisons were made against data captured during a long term 12 month study. The findings indicate that the suggested approach to human behaviour simulation creates sufficiently similar results, in terms of agents

paths, to enable the creation of accurate interaction maps for risk assessment. Accuracy was assessed through the use of histogram comparison, demonstrating logical similarity between simulated and real paths from the same environment. Visibility maps based on the same simulation algorithm were generated to enable assessment of an area on both interaction and visibility, better emulating the way a human would interact with their environment. Replication of a human decision making ability to re plan their route on discovery of a risk was also incorporated, demonstrating that interaction works both ways with the environment also having an impact on how a human uses the space.

A Human and Group Behaviour Simulation framework was also presented with a number of key benefits over existing methods. As the framework only takes in a source video as an input, a number of the time consuming ground truth definition and annotation steps are reduced. A source video does not require extensive preprocessing to accurately determine number of agents in a scene or their recorded tracks through the sequence. This avoids a lengthy manual or semiautomated process. The framework also allows researchers who wish to compare their algorithm against others a quick and efficient way of doing so, either by using the same well-known source material and datasets in the field or simply rerunning the framework with other pedestrian or crowd simulation algorithms to compare with. Additionally for model tuning; the proposed method can create a fast feedback loop that allows the modification of parameters to improve simulation accuracy. As the ground truth data for any simulated visualisations is already intrinsically known, and as specific testing scenarios and behaviours can be simulated, the methodology is also very suitable for the evaluation of pedestrian tracking algorithms on video data.

The Human and Group Behaviour Simulation framework reduces the complexity of simulation evaluation and provides tangible and relevant metrics that can be used for comparison and parametric tuning. A perspective plane extraction process is introduced which allows the conversion of source material into a simulated/composited video with controlled agents (3D models) replacing those of the humans. Through the use of a modular system, any crowd or pedestrian simulation model can be evaluated and compared by generating agent motion for use in the final visual simulation. Features utilising Weber's Law, with regards to vision, are utilised to replicate the Human Visual System's ability to perceive movement. Through evaluation on a large range of challenging and diverse scenes, it has been shown that the methodology presents quantifiable measures

of video properties such as speed and number of agents. In addition the combined effect of these features correlate well with human participant analysis of the same videos showing that the system closely emulates the Human Visual System.

5.5 Discussion

Using the concepts of environmental risk as well as those mentioned in the precious two chapters, a reasonably comprehensive view of domestic risk is achieved. These presented concepts are deliberately broad, so as to allow for a wider range of applications as possible. However the presented Risk Estimation framework is also designed to be customisable. The main focus of this presented work is that of domestic environments, however with small modifications to the weightings the system can be tailored to other environments and tasks, for example the previously given lab scenario, where interaction analysis and stability assessment would be prioritized over the hazard properties of objects in the scene.

Further discussions and conclusions are given in the following chapter where the future of this research area is addressed, along with an analysis of the work to date and the logical extensions.

Chapter 6

Conclusion

6.1 Conclusions and Future Work

The task of automated risk assessment in a domestic environment is one that has received little dedicated research to date. The issues surrounding the identification, classification and, finally, quantification of risk are substantial. Practical issues of identifying a risk or hazard, as well as the contextual problem of defining what is considered as hazardous and what is not, are both non trivial tasks. As such this thesis has aimed to tackle these issues with a number of advancements focused within a domestic setting. An outline of the individual problems and their associated issues is given below, along with the proposed solutions presented within this thesis and expected future work.

6.2 Stability Assessment

In an effort to determine whether an object within a scene is in a hazardous position, a method to analyse that objects stability is required. This provides the ability, through the use of physics simulation techniques, to highlight those objects considered to be in an unstable position and define whether or not the placement of an object in a scene presents a potential risk. Within the confines of the domestic setting this analysis helps create a preventative, rather than reactive, system, allowing action to be taken before a hazardous situation develops. This is especially important for those at risk users whose ability to determine risk in their environment may be diminished or under developed.

6.2.1 Issues

Existing techniques for stability assessment involve the application of a probabilistic model to determine the force required to dislodge a particular object [16]. This results in a local assessment of the object and application force, with no consideration to the wider scene and the knock on effects that the object might have. Given the domestic nature of the problem, any suitable methodology must consider the likely computational power available. Additionally the lack of a dedicated risk dataset makes comparison of stability estimation techniques difficult.

6.2.2 Proposed Solution

A novel predictive physics based stability analysis technique is introduced allowing the quantification of instability for a scene [15, 34]. Consideration is given to the effect that the movement of an object will have on the rest of the scene, utilising simulation techniques. Through the use of this system a detailed picture is created of how the application of forces will effect the scene. Importantly the concept of reinforced stability through the presence of other objects is displayed in this method, with the results demonstrating that with objects placed closer together and further towards the centre of the a table the instability of a scene decreases.

To allow for the increased computational requirement of a full simulation process, a regression based prediction method is implemented [30, 31] providing measured energy outputs and stability assessments without the need for full simulation. However the produced risk scores as a result of the prediction framework, do not follow the same conclusions as the full simulation. This illustrates a need to better model these complex object movements and interactions to provide a more applicable risk score.

A dedicated risk dataset in the form of the 3D Risk Scenes (3DRS) dataset is also presented allowing risk related methodologies to be tested within a standardised environment. This allows for the testing and evaluating of risk based methodologies on standardised scenarios, allowing the comparison of outputs between various methods. As risk evaluation is a contextual issue, the comparison of specific numeric results between methodologies may not be informative, however the demonstration of risk trends, such as those displayed in the stability analysis, would make a useful evaluation technique.

6.2.3 Future Work

A more refined regression modelling technique is required, which would result in fewer, more general models and would enable a more reactive and faster system. With the advent of more specific embedded technologies (such as Nvidia's Tegra Chipset), the ability to do complex simulation in a domestic environment is becoming more accessible. However given the amount of simulations required to produce a risk score for the given 3DRS scenarios, prediction would be a preferable option.

To help improve simulation accuracy experimentation with more complex bounding shapes for object models could be implemented, this would help provide more accurate energy predictions but would increase computation time and add additional steps to the processing pipeline. Additionally extension to to the 3DRS dataset to include a more diverse set of risk scenes would help in the development of a more robust and applicable system.

6.3 Hazard Feature Recognition

To form a more comprehensive view of risk within a scene, a method is required which is able to define whether any objects present within the environment may pose a hazard. This additional information about the scene allows any stability estimation method a further degree of relevance by adding further context around any object in an unstable position. The definition of whether an object is hazardous or not inevitably leads to the problem of what is considered hazardous. As well as to whom. Both considerations which are not well addressed in existing research.

6.3.1 Issues

Object risk definition though recognition is impractical, primarily due to the problems of training a robust model by which all household objects could be classified. Object recognition also presents the problem of similar object types having different levels of risk. With the intended domestic application, thought must be given to the likely available hardware and as such any new risk detection system should make use of existing data where possible aiding in computational efficiency. As with any classification task, model

training times and avoidance of over fitting the model to the data must be considered all whilst still retaining good recognition accuracy on new unseen test samples.

6.3.2 Proposed Solution

To address the task of hazard feature recognition, a number of 3D feature descriptors are presented which aim to identify hazardous properties of an object, avoiding the more problematic task of object recognition. The 3D Voxel HOG descriptor [15, 32] allows the definition of a model for object properties such as sharp edges or corners. Additionally the Physics Behaviour Feature [29] is introduced which reuses simulation data to define risk by encoding the way objects react to applied forces as a feature vector. When combined with 3D VHOG and trained using Adaboost, this produces a highly sensitive classification model for safe and unsafe objects, able to highlight all hazardous objects in the 3DRS dataset, with a high F1 score (0.750).

A robust filtering process [34] is also suggested, increasing the robustness of the feature descriptors to further improve classification accuracy on a range of computer vision tasks. Finally a complex variant of Adaboost [34] is suggested and evaluated which reduces the training time and number of iterations of classification models and takes advantage of the intrinsic relationship of complex and hyper complex numbers. With the addition of these methods to the 3D VHOG and Physics behaviour features the overall F1 score is increased to 0.828, whilst maintaining the same sensitivity to hazardous objects that is required for this type of application.

6.3.3 Future Work

A logical next step to the framework is an extension to identifiable risks, expanding the detection mechanism to include other types of hazardous object properties. This requires further analysis on what is considered risky and would help define a system that is more suitable for the domestic environment. The system would also benefit from evaluation on a larger dataset of domestic objects to improve applicability to the task. Currently, although the hazard feature recognition is high, the limited set of objects in the dataset does not provide a broad enough view of the domestic setting. Additionally consideration should be given to the suitability of other machine learning techniques.

Further analysis of the properties of the proposed 3D VHOG should also be investigated, as the presented low level structure analysis may have applications in other sectors of research.

6.4 Environmental Risk and Human Behaviour Simulation

The effect that human interaction has on detected risks in an environment requires further study. Conversely the effect those risks have on the humans within that environment must also be considered. More importantly a method is required to predict these effects and therefore provide further insight into those detected risks, allowing for a more complete picture of risk in an environment.

6.4.1 Issues

The definition of simulation algorithms and models for predicting human behaviour is a challenge, primarily due to the need for careful consideration when defining which aspects of human behaviour to model. This issue is exacerbated when considering the problem of modeling behaviour in the presence of risk. Additionally this simulation data then requires conversion into a quantifiable risk score based on human interaction with an environment. Finally the definition of simulation accuracy must be considered to ensure that the produced behaviour is realistic.

6.4.2 Proposed Solution

Environmental risk maps, which quantify human presence and environmental interaction, are introduced to produce an element for use in the Risk Estimation framework. Environmental maps utilise simulation techniques based on two concepts: environment visibility, to assess a human's ability to *see* potential risks and path analysis, to simulate the areas of the scene most likely to be visited.

Environmental risk maps rely on the accuracy of the simulation algorithm used to produce realistic human behaviour in an environment. This forms the basis from which the interaction and visibility components of the environmental risk maps are calculated, ensuring that outputted risk scores are logical and relevant. To further improve human

behaviour emulation a risk behaviour simulation algorithm is introduced based on the expected utility for an agent's action, helping to model a human's behaviour in the presence of risk. Comparing the paths and interactions produced from the simulation algorithm to those produced by human participants validates the method; highlighting a the link between accurate human behaviour simulation and realistic risk scores.

Finally as a form of validation, an evaluation framework is presented which analyses how realistic a simulation algorithm reproduces human behaviour through the replication of a human's ability to identify similarities in movement. The proposed framework allows the comparison between simulated videos and source footage, allowing for the tuning of simulation parameters or comparisons between different algorithms entirely providing a simplified form of evaluation without the need for complex groundtruth steps. Evaluation was carried out using a group of ten participants asked to grade a set of simulated videos against source footage. The framework provided similar scores when evaluating the same videos against the source, highlighting the frameworks ability to deduce changes in the crowd.

6.4.3 Future Work

Extension of the human behaviour simulation algorithm is required to allow the consideration of more elements in the decision making process, this would allow for a more tailored approach to the simulation of agents in a scene, allowing emulation of humans with specific disabilities or limitations. This requires a detailed analysis of which factors to be considered and how applicable they are. The establishment of a risk based dataset for human behaviour would be a significant contribution to the research field, due to the difficulty in acquiring this type of data. A domestic focus to this would be able to provide a dataset of domestic incidents from which evaluation of detection techniques could be validated. This would be particularly useful for the wider human behaviour simulation research community, given that many of these techniques are utilised in disaster and evacuation simulation and modelling.

6.5 Epilogue

The goal of this thesis was to make advancements into the relatively new field of automated risk assessment for domestic scenes. This was undertaken from a number of angles; novel 3D descriptors for hazardous object property recognition were introduced, a robust and more applicable form of stability prediction was implemented and finally more in depth analysis on human interaction with regards to risk was considered.

Domestic robotics and smart homes are a growing industry and will become an integral part of life in the near future. Currently available commercial products aim to perform menial tasks, simplifying processes that humans perform every day. For example taking notes, the initialisation of domestic appliances, information retrieval and simple household chores. With the ever more interconnected nature of the domestic environment and the accessability of increasing computational power, these devices will take on new roles. The ability to provide basic decision making capabilities as well as more detailed interaction and analytical abilities will enable the application of more complex behaviour.

In a domestic setting this will likely lead to the performing of more complex, but still rudimentary, tasks such as assisting with more complex chores, heavy lifting and entertainment. These abilities will be determined by the developed hardware, and in the near future will likely focus on individual tasks as opposed to a one size fits all solution. However for tasks involving the presence or potentially the monitoring of those that use the environment (e.g. children or at risk adults), elements of risk detection will be required to ensure user safety. These concepts will be required in the emulation of further higher level behaviours and will produce an initial step in the development of systems that can make a real difference in easing some of the impending social issues to do with health care and aging populations.

Bibliography

[1] O. f. N. Statistics, "2011 Census: Population and household estimates for the United Kingdom," Tech. Rep., 2012. [Online]. Available: http://webarchive.nationalarchives.gov.uk/20160105160709/http://www.ons.gov.uk/ons/guide-method/census/2011/census-data/2011-census-data-catalogue/population-and-household-estimates/index.html

[2] MrReid, "The physics of the Kinect," 2014. [Online]. Available: http://wordpress.mrreid.org/2011/08/20/kinect-physics/

[3] Autodesk, "Point cloud object," 2014. [Online]. Available: http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Help/index.html?url=files/GUID-49CE0ACB-1345-4D50-B6E5-361DBFDB5B33.htm,topicNumber=d30e158270

[4] S. Izadi, A. Davison, A. Fitzgibbon, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, and D. Freeman, "Kinect Fusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User Interface Software and Technology*, 2011, pp. 559–568.

[5] B. Zheng, Y. Zhao, J. C. Yu, K. Ikeuchi, and S. C. Zhu, "Beyond point clouds: Scene understanding by reasoning geometry and physics," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3127–3134, jun 2013.

[6] M. Pharr and R. Fernando, *Gpu gems 2: programming techniques for high-performance graphics and general-purpose computation.* Addison-Wesley Professional, 2005.

[7] Miyata Cycle Company Ltd, "Miyata," 2014. [Online]. Available: https://cyclingiq.com/2012/01/26/miyata-japanese-road-bicycle-legend-reborn/

[8] W. J. Jeon, G. A. R. Sanchez, T. Lee, Y. Choi, B. Woo, K. Lim, and H. Byun, "Real-time detection of speed-limit traffic signs on the real road using Haar-like features and boosted cascade," *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication - ICUIMC '14*, pp. 1–5, 2014.

[9] C. Prall, "A comparison of javascript physics engines," 2012. [Online]. Available: http://www.webappers.com/2012/12/11/a-comparison-of-javascript-physics-engines/

[10] C. W. Reynolds, "Steering behaviors for autonomous characters," in *Game Developers Conference*, vol. 1999, 1999, pp. 763–782.

[11] B. Zheng, Y. Zhao, J. Yu, K. Ikeuchi, and S. C. Zhu, "Scene understanding by reasoning stability and safety," *International Journal of Computer Vision*, vol. 112, no. 2, pp. 221–238, 2015.

[12] L. Yin, X. Wei, Y. Sun, J. Wang, and M. Rosato, "A 3D facial expression database for facial behavior research," *7th International Conference on Automatic Face and Gesture Recognition*, vol. 10, no. 12, pp. 211–216, 2006.

[13] J. Ferryman and A. Ellis, "PETS2010: Dataset and challenge," *Proceedings - IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2010*, pp. 143–148, 2010.

[14] E. I. Konstantinidis, A. S. Billis, L. Plotegher, G. Conti, and P. D. Bamidis, "Indoor location IoT analytics "in the wild": Active and healthy ageing cases," in *XIV Mediterranean Conference on Medical and Biological Engineering and Computing 2016*. Springer, 2016, pp. 1225–1230.

[15] R. Dupre, V. Argyriou, and D. Greenhill, "A 3D Scene Analysis Framework and Descriptors for Risk Evaluation," in *International Conference on 3D Vision (3DV)*. IEEE, 2015, pp. 100–108.

[16] B. Zheng, Y. Zhao, J. C. Yu, K. Ikeuchi, and S. C. Zhu, "Detecting potential falling objects by inferring human action and natural disturbance," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2014, pp. 3417–3424.

[17] M. Scherer, M. Walter, and T. Schreck, "Histograms of oriented gradients for 3d object retrieval," in *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2010, pp. 41–48.

[18] R. B. Fisher, *From surfaces to objects: Computer vision and three dimensional scene analysis*. John Wiley and Sons, 1989.

[19] L. G. Roberts, "Machine perception of three-dimensional solids," Ph.D. dissertation, 1965.

[20] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, jun 2011, pp. 1297–1304.

[21] Office for National Statistics, "Disability in England and Wales: 2011 and comparison with 2001," 2013. [Online]. Available: http://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/disability/articles/disabilityinenglandandwales/2013-01-30

[22] The Royal Society for the Prevention of Accidents, "HASS and LASS data," Tech. Rep., 2002. [Online]. Available: http://www.hassandlass.org.uk/reports/2002data.pdf{%}5Cnhttp://www.hassandlass.org.uk/query/MainSelector.aspx

[23] J. Melorose, R. Perroy, and S. Careas, "World population prospects," Tech. Rep., 2015. [Online]. Available: https://esa.un.org/unpd/wpp/publications/files/key{_}findings{_}wpp{_}2015.pdf

[24] M. Chan, D. Estève, C. Escriba, and E. Campo, "A review of smart homes-Present state and future challenges," *Computer Methods and Programs in Biomedicine*, vol. 91, no. 1, pp. 55–81, 2008.

[25] T. S. Tadele, T. De Vries, and S. Stramigioli, "The safety of domestic robotics: A survey of various safety-related publications," *IEEE Robotics and Automation Magazine*, vol. 21, no. 3, pp. 134–142, 2014.

[26] International Federation of Robotics, "World 2014 robotics survey - executive summary," *World Robotic Report - Executive Summary*, pp. 10–21, 2014.

[27] C. A. Smarr, T. L. Mitzner, J. M. Beer, A. Prakash, T. L. Chen, C. C. Kemp, and W. A. Rogers, "Domestic robots for older adults: Attitudes, preferences, and potential," *International Journal of Social Robotics*, vol. 6, no. 2, pp. 229–247, 2014.

[28] C. Harper and G. Virk, "Towards the development of international safety standards for human robot interaction," *International Journal of Social Robotics*, vol. 2, no. 3, pp. 229–234, 2010.

[29] R. Dupre and V. Argyriou, "3D Voxel HOG and Risk Estimation," in *International Conference on Digital Signal Processing, DSP*, vol. 2015-Septe, 2015, pp. 482–486.

[30] R. Dupre, V. Argyriou, and D. Greenhill, "Prediction of Physics Simulations for Graphics and Animation," *Sigrad 2014*, 2014.

[31] R. Dupre and V. Argyriou, "Prediction of physics simulations for graphics and animation," in *Proceedings of SIGRAD 2014, Visual Computing, June 12-13, 2014, G{ö}teborg, Sweden*, no. 106. Link{ö}ping University Electronic Press, 2014, pp. 83–86.

[32] R. Dupre, V. Argyriou, and D. Greenhill, "Risk assessment for RGBD scans in real time," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2016-May, 2016, pp. 2084–2088.

[33] Y. Freund and R. Schapire, "A Desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational Learning Theory*, 1995, pp. 23–37.

[34] R. Dupre, V. Argyriou, G. Tzimiropoulos, and D. Greenhill, "Risk analysis for smart homes and domestic robots using robust shape and physics descriptors , and complex boosting techniques," *Information Sciences*, vol. 372, pp. 359–379, 2016.

[35] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *European Conference Computer Vision*, 2004, pp. 224–237.

[36] J. Niemeyer, F. Rottensteiner, and U. Soergel, "Contextual classification of lidar data and building object detection in urban areas," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 87, no. 1, pp. 152–165, 2014.

[37] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet, "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 105, no. 7, pp. 286–304, 2015.

[38] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald, "Real-time large-scale dense RGB-D SLAM with volumetric fusion," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 598–626, 2015.

[39] J. Wang, C. Zhang, W. Zhu, Z. Zhang, Z. Xiong, and P. A. Chou, "3D scene reconstruction by multiple structured-light based commodity depth cameras," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 5429–5432.

[40] J. Huang, R. Yagel, V. Filippov, and Y. Kurzion, "An accurate method for voxelizing polygon meshes," in *IEEE Symposium on Volume Visualization.* Ieee, 1998, pp. 119–126.

[41] A. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, "Efficient organized point cloud segmentation with connected components," in *Proceedings of Semantic Perception Mapping and Exploration*, 2013, pp. 1–6.

[42] Z. Jia, A. Gallagher, A. Saxena, and T. Chen, "3D-Based reasoning with blocks, support, and stability," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, jun 2013.

[43] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, "Exploring features in a Bayesian framework for material recognition," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 239–246, jun 2010.

[44] C. Liu, G. Yang, and J. Gu, "Learning discriminative illumination and filters for raw material classification with pptimal projections of bidirectional texture functions," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1430–1437, jun 2013.

[45] M. a. Mofaddel and W. M. Abd-Elhafiez, "Fast and accurate approaches for image and moving object segmentation," *The 2011 International Conference on Computer Engineering & Systems*, pp. 252–259, nov 2011.

[46] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 3241–3248, nov 2011.

[47] D. Fox, "RGB-(D) scene labeling: Features and algorithms," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, no. D, pp. 2759–2766, jun 2012.

[48] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, jun 2010.

[49] X. Liu, S. Cheng, X. Zhang, X. Yang, T. B. Nguyen, and S. Lee, "Unsupervised segmentation in 3D planar object maps based on fuzzy clustering," *2012 Eighth International Conference on Computational Intelligence and Security*, pp. 364–368, nov 2012.

[50] C. Do and B. Javidi, "3D Integral imaging reconstruction of occluded objects using independent component analysis-based K-means clustering," *Journal of Display Technology*, vol. 6, no. 7, pp. 257–262, 2010.

[51] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum, "Simulation as an engine of physical scene understanding." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110, no. 45, pp. 18 327–32, nov 2013.

[52] J. Wu, I. Yildirim, J. Lim, W. Freeman, and J. Tenenbaum, "Galileo : Perceiving physical object properties by integrating a physics engine with deep learning," in *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015, pp. 1–9.

[53] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: an efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, feb 2013.

[54] G. Kou, Y. Peng, and G. Wang, "Evaluation of clustering algorithms for financial risk analysis using MCDM methods," *Information Sciences*, vol. 275, pp. 1–12, 2014.

[55] E. Stone and M. Skubic, "Evaluation of an inexpensive depth camera for passive in-home fall risk assessment," in *Proceedings of International ICST Conference on Pervasive Computing Technologies for Healthcare*. Ieee, 2011, pp. 71–77.

[56] A. N. Belbachir, A. Nowakowska, S. Schraml, G. Wiesmann, and R. Sablatnig, "Event-driven feature analysis in a 4D spatiotemporal representation for ambient assisted living," *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 1570–1577, nov 2011.

[57] T. Dannenmann, "Novel safety feature to protect critical anatomical structures during navigation-guided robotic surgery," in *Jahrestagung der Deutschen Gesellschaft fur Computer-und Roboterassistierte Chirurgie, CURAC 2*, 2003.

[58] B. Dhillon, A. Fashandi, and K. Liu, "Robot systems reliability and safety: a review," *Journal of Quality in Maintenance Engineering*, vol. 8, no. 3, pp. 170–212, 2002.

[59] Department of Labour. Industrial Welfare Division. New Zealand, *Robot Safety*. Industrial Welfare Division, Department of Labour, 1987.

[60] C. Sharp, O. Shakernia, and S. Sastry, "A vision system for landing an unmanned aerial vehicle," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2. Ieee, 2001, pp. 1720–1727.

[61] K. E. Wenzel, A. Masselli, and A. Zell, "Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 221–238, oct 2010.

[62] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3D objects," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 263–284, 2007.

[63] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings - IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. I, 2005, pp. 886–893.

[64] D. Lowe, "Object recognition from local scale-invariant features," in *IEEE Conference on Computer Vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[65] Y. Sun, L. Zhao, S. Huang, L. Yan, and G. Dissanayake, "L2-SIFT: SIFT feature extraction and matching for large images in large-scale aerial photogrammetry," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. Volume 91, pp. 1–16, 2014.

[66] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," *Sixth International Conference on Computer Vision*, pp. 555–562, 1998.

[67] P. Felzenswalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, sep 2010.

[68] N. Buch, M. Cracknell, J. Orwell, and S. Velastin, "Vehicle localisation and classification in urban CCTV streams," *16th World Congress and Exhibition on Intelligent Transport Systems and Services*, pp. 1–8, 2009.

[69] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao, "Histogram of oriented normal vectors for object recognition with a depth sensor," in *Proceedings of Asian conference on Computer Vision*, vol. 2, 2012, pp. 525–538.

[70] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman, "Human focused action localization in video," in *Trends and Topics in Computer Vision*, vol. 6553 LNCS, 2012, pp. 219–233.

[71] A. Prest, V. Ferrari, and C. Schmid, "Explicit modeling of human-object interactions in realistic videos." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 835–48, apr 2013.

[72] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *European Conference Computer Vision*, 2010, pp. 356–369.

[73] P. Cirujeda, Y. Dicente Cid, X. Mateo, and X. Binefa, "A 3D scene registration method via covariance descriptors and an evolutionary stable strategy game theory solver: fusing photometric and shape-based features," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 306–329, 2015.

[74] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217.

[75] A. Flint, A. Dick, and A. Van Den Hengel, "Thrift: Local 3D structure recognition," in *Digital Image Computing Techniques and Applications: 9th Biennial Conference of the Australian Pattern Recognition Society*, 2007, pp. 182–188.

[76] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: efficient and robust 3D object recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, jun 2010, pp. 998–1005.

[77] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[78] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances In Neural Information Processing Systems*, pp. 1097–1105, 2012.

[79] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[80] T. M. Mitchell, "Machine Learning." *Burr Ridge, IL: McGraw Hill*, vol. 45, p. 14, 1997.

[81] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, sep 1995.

[82] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances In Neural Information Processing Systems*, vol. 9, pp. 155–161, 1997.

[83] J. Wu, L. Huang, and X. Pan, "A novel bayesian additive regression trees ensemble model based on linear regression and nonlinear regression for torrential rain forecasting," *2010 Third International Joint Conference on Computational Science and Optimization*, pp. 466–470, 2010.

[84] J. Boyle, M. Wallis, M. Jessup, J. Crilly, J. Lind, P. Miller, and G. Fitzgerald, "Regression forecasting of patient admission data." in *IEEE Engineering in Medicine and Biology Society. Conference*, jan 2008, pp. 3819–22.

[85] F. Luo, C. Liu, and Z. Sun, "Intelligent vehicle simulation and debugging environment based on physics engine," *International Asia Conference on Informatics in Control, Automation and Robotics*, pp. 329–333, feb 2009.

[86] H. Lu and W. Yijin, "Design and implementation of three-dimensional game engine," *World Automation Congress (WAC)*, pp. 1–4, 2012.

[87] D. C. C. Peixoto, R. M. Possa, R. F. Resende, and C. I. P. S. Pádua, "An overview of the main design characteristics of simulation games in Software Engineering education," in *2011 24th IEEE-CS Conference on Software Engineering Education and Training*, 2011, pp. 101–110.

[88] H. Gould, J. Tobochnik, and W. Christian, *Introduction to computer simulation methods: Application to physical systems*, 3rd ed. Addison-Wesley, 2006.

[89] V. Karamain, *Introduction to Game Programming: Using C# and Unity 3D.* Noorcon, 2016.

[90] D. Baraff, "Non-penetrating rigid body simulation," *State of the Art Reports*, 1993.

[91] K. Egan, "Techniques for real-time rigid body simulation," Ph.D. dissertation, Brown University, 2003.

[92] C. Delgado-Mata and J. Ib'nez, "Adaptive physics for game-balancing in videogames for social interaction," *2011 International Conference on Technologies and Applications of Artificial Intelligence*, pp. 254–259, nov 2011.

[93] D. F. Silva and A. Maciel, "A comparative study of physics engines for modeling soft tissue deformation," *2012 XXXVIII Conferencia Latinoamericana En Informatica (CLEI)*, pp. 1–7, oct 2012.

[94] a. Roennau, F. Sutter, G. Heppner, J. Oberlaender, and R. Dillmann, "Evaluation of physics engines for robotic simulations with a special focus on the dynamics of walking robots," *2013 16th International Conference on Advanced Robotics (ICAR)*, pp. 1–7, nov 2013.

[95] M. Asano, T. Iryo, and M. Kuwahara, "A pedestrian model considering anticipatory behaviour for capacity evaluation," *Transportation and Traffic Theory*, vol. 18, p. 28, 2009.

[96] F. Klugl, G. Klubertanz, and G. Rindsfuser, "Agent-based pedestrian simulation of train evacuation integrating environmental data," in *Lecture Notes in Computer Science*, vol. 5803, 2009, pp. 631–638.

[97] H. Xi, S. Lee, and Y.-j. Son, "An integrated pedestrian behavior model based on extended decision field theory and social force model," in *Human-in-the-Loop Simulations*, 2011, pp. 69–95.

[98] H. Jiang, W. Xu, T. Mao, C. Li, S. Xia, and Z. Wang, "Continuum crowd simulation in complex environments," *Computers & Graphics*, vol. 34, no. 5, pp. 537–544, 2010.

[99] A. Treuille, S. Cooper, and Z. Popović, "Continuum crowds," *ACM Transactions on Graphics*, vol. 25, no. 3, p. 1160, 2006.

[100] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4. ACM, 1987, pp. 25–34.

[101] D. Helbing and P. Molnar, "Self-organization phenomena in pedestrian crowds," *Condensed Matter*, pp. 569–577, 1998.

[102] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[103] I. Karamouzas, P. Heil, P. Van Beek, and M. H. Overmars, "A predictive collision avoidance model for pedestrian simulation," *Lecture Notes in Computer Science*, vol. 5884, pp. 41–52, 2009.

[104] S. Stroeve, H. Blom, and M. van der Park, "Multi-agent situation awareness error evolution in accident risk modelling," *5th USA/Europe Air Traffic Management R&D Seminar*, no. June, pp. 23–27, 2003.

[105] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 32–64, 1995.

[106] S. Kim, S. J. Guy, D. Manocha, and M. C. Lin, "Interactive simulation of dynamic crowd behaviors using general adaptation syndrome theory," *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games - I3D*, vol. 1, no. 212, p. 55, 2012.

[107] D. Crompton, "Pedestrian delay, annoyance and risk: pre-liminary results from a 2 years study," in *In Proceedings of PTRC Summer Annual Meeting*, 1979, pp. 275–299.

[108] D. C. Duives, W. Daamen, and S. P. Hoogendoorn, "State-of-the-art crowd motion simulation models," *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 193–209, 2013.

[109] E. Papadimitriou, G. Yannis, and J. Golias, "A critical assessment of pedestrian behaviour models," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 12, no. 3, pp. 242–255, 2009.

[110] A. Portz and A. Seyfried, "Analyzing stop-and-go waves by experiment and modeling," in *Pedestrian and Evacuation Dynamics*, 2010, pp. 577–586.

[111] M. Asano, T. Iryo, and M. Kuwahara, "Microscopic pedestrian simulation model combined with a tactical model for route choice behaviour," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 6, pp. 842–855, 2010.

[112] Q. Wang, Y. Liu, and J. Chen, "Accurate indoor tracking using a mobile phone and non-overlapping camera sensor networks," *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pp. 2022–2027, 2012.

[113] A. Lerner, Y. Chrysanthou, A. Shamir, and D. Cohen-Or, "Data driven evaluation of crowds," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5884 LNCS, pp. 75–83, 2009.

[114] A. Lerner, Y. Chrysanthou, and A. Shamir, "Context-dependent crowd evaluation," *Computer Graphics Forum*, vol. 29, no. 7, pp. 2197–2206, 2010.

[115] S. Munder, C. Schnörr, and D. M. Gavrila, "Pedestrian detection and tracking using a mixture of view-based shape – texture models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 2, pp. 333–343, 2008.

[116] M. Raptis and S. Soatto, "Tracklet descriptors for action modeling and video analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6311 LNCS, no. PART 1, pp. 577–590, 2010.

[117] P. Allain, N. Courty, and T. Corpetti, "Crowd flow characterization with optimal control theory," *Asian Conference on Computer Vision (ACCV)*, pp. 279–290, 2009.

[118] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 34, no. 3, pp. 334–352, 2004.

[119] T. I. Lakoba, D. J. Kaup, and N. M. Finkelstein, "Modifications of the Helbing-Molnár- Farkas- Vicsek social force model for pedestrian evolution," *Simulation*, vol. 81, no. 5, pp. 339–362, 2005.

[120] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[121] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *7th International Joint Conference on Artificial intelligence*, vol. 2, 1981, pp. 674–679.

[122] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2432–2439, 2010.

[123] D. Sun, S. Roth, and M. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 115–137, 2013.

[124] T. L. Clarke, D. Kaup, L. Malone, R. Oleson, and M. Rosa, "Crowd model verification using video data," *Proceedings of EMSS 2007*, pp. 4–6, 2007.

[125] P. Charalambous, I. Karamouzas, S. Guy, and Y. Chrysanthou, "A data-driven framework for visual crowd analysis," *Computer Graphics Forum*, vol. 33, no. 7, pp. 41–50, 2014.

[126] S. J. Guy, J. van den Berg, W. Liu, R. Lau, M. C. Lin, and D. Manocha, "A statistical similarity measure for aggregate crowd dynamics," *ACM Transactions on Graphics*, vol. 31, no. 6, p. 1, 2012.

[127] M. Kapadia, M. Wang, S. Singh, G. Reinman, and P. Faloutsos, "Scenario space: Characterizing coverage, quality, and failure of steering algorithms," *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '11*, vol. 1, p. 53, 2011.

[128] M. Rodriguez, J. Sivic, I. Laptev, and J.-Y. Audibert, "Data-driven crowd analysis in videos," in *International Conference on Computer Vision*, 2011, pp. 1235–1242.

[129] J. Pettré, J. Ondrej, A.-h. Olivier, A. Cretual, and S. Donikian, "Experiment-based modeling, simulation and validation of interactions between virtual walkers," *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, vol. 2009, p. 189, 2009.

[130] H. Wang, J. Ondřej, and C. O'Sullivan, "Path patterns: Analyzing and comparing real and simulated crowds," in *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games - I3D '16*, no. February, 2016, pp. 49–57.

[131] S. R. Musse, V. J. Cassol, and C. R. Jung, "Towards a quantitative approach for comparing crowds," *Computer Animation and Virtual Worlds*, vol. 23, no. 1, pp. 49–57, 2012.

[132] K. Jablonski, V. Argyriou, D. Greenhill, and S. A. Velastin, "Evaluation framework for crowd behaviour simulation and analysis based on real videos and scene reconstruction," in *IET The 6th Latin American Conference on Networked and Electronic Media LACNEM*, 2015.

[133] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," *European Conference Computer Vision*, pp. 1–14, 2012.

[134] C. C. Loy, S. Gong, and T. Xiang, "From semi-supervised to transfer counting of crowds," *2013 IEEE International Conference on Computer Vision*, pp. 2256–2263, 2013.

[135] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting," *Procedings of the British Machine Vision Conference 2012*, vol. 1, no. 2, pp. 1–11, 2012.

[136] D. Simonnet, S. A. Velastin, J. Orwell, and E. Turkbeyler, "Selecting and evaluating data for training a pedestrian detector for crowded conditions," *2011 IEEE International Conference on Signal and Image Processing Applications, ICSIPA 2011*, pp. 174–179, 2011.

[137] W. Hu, Z. Qu, and X. Zhang, "A new approach of mechanics simulation based on game engine," *Computational Sciences and Optimization (CSO), 2012 Fifth International Joint Conference on. IEEE*, 2012.

[138] M. Servin, C. Lacoursiere, and N. Melin, "Interactive simulation of elastic deformable materials," *Proceedings of SIGRAD Conference (2006)*, pp. 22–32, 2006.

[139] B.-S. Kim, P. Kohli, and S. Savarese, "3D Scene understanding by voxel-CRF," in *IEEE International Conference on Computer Vision*. Ieee, dec 2013, pp. 1425–1432.

[140] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.

[141] O. Wang, P. Gunawardane, S. Scher, and J. Davis, "Material classification using BRDF slices," in *IEEE Conference on Computer Vision and Pattern Recognition*, jun 2009, pp. 2805–2811.

[142] Y. Kobayashi, T. Morimoto, I. Sato, Y. Mukaigawa, and K. Ikeuchi, "BRDF Estimation of structural color object by using hyper spectral image," in *IEEE International Conference on Computer Vision Workshop*, dec 2013, pp. 915–922.

[143] A. Davis, K. L. Bouman, J. G. Chen, M. Rubinstein, F. Durand, and W. T. Freeman, "Visual vibrometry: Estimating material properties from small motion

in video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5335–5343.

[144] D. Casanova, J. Florindo, M. Falvo, and O. Bruno, "Texture analysis using fractal descriptors estimated by the mutual interference of color channels," *Information Sciences*, vol. 346, no. 10, pp. 58–72, feb 2016.

[145] Real-Time Physics Simulation, "Bullet user manual and API documentation," 2012. [Online]. Available: http://bulletphysics.org/mediawiki-1.5.8/index.php/Bullet{_}User{_}Manual{_}and{_}API{_}documentation

[146] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings International Conference on Multimedia*. New York, New York, USA: ACM Press, 2007, pp. 357–360.

[147] A. Godil and A. Wagan, "Salient local 3D features for 3D shape retrieval," in *IS&T/SPIE Electronic Imaging*, 2011, pp. 78 640S—-78 640S.

[148] I. Sipiran and B. Bustos, "Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes," *The Visual Computer*, vol. 27, no. 11, pp. 963–976, jul 2011.

[149] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010.

[150] A. J. Fitch, A. Kadyrov, W. J. Christmas, and J. Kittler, "Fast robust correlation," *IEEE Transactions on Image Processing*, vol. 14, no. 8, pp. 1063–1073, 2005.

[151] S. Liwicki, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Euler principal component analysis," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 498–518, 2012.

[152] J. H. Morra, Z. Tu, L. G. Apostolova, A. E. Green, A. W. Toga, and P. M. Thompson, "Comparison of AdaBoost and support vector machines for detecting Alzheimer's disease through automated hippocampal segmentation." *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 30–43, jan 2010.

[153] T. Adali, P. J. Schreier, and L. L. Scharf, "Complex-valued signal processing: The proper way to deal with impropriety," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5101–5125, 2011.

[154] X. L. Li and T. Adali, "Noncircular Principal Component Analysis and Its Application to Model Selection," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4516–4528, 2011.

[155] N. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," *Advances in Neural Information Processing Systems*, vol. 16, no. 3, pp. 329–336, 2004.

[156] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition*, dec 2001, pp. 329–336.

[157] B.Han, C. Yang, R. Duraiswami, and L. Davis, "Bayesian filtering and integral image for visual tracking," in *WIAMIS*, 2005, pp. 329–336.

[158] E. Tapia, "A note on the computation of high-dimensional integral images," *Pattern Recognition Letters*, vol. 32, no. 2, pp. 197–201, 2011.

[159] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in *IEEE Conference on Computer Vision*, vol. 1, 2005, pp. 166–173.

[160] E. U. Weber, "De Pulsu, Resorptione, Auditu et Tactu," *Annotationes anatomicae et physiologicae*, pp. 44–174, 1834.

[161] E. Wharton, K. Panetta, and S. Agaian, "Human visual system based similarity metrics," in *IEEE International Conference on Systems, Man and Cybernetics*, 2008, pp. 685–690.

[162] J. M. Zanker, "Does motion perception follow Weber's law?" *Perception*, vol. 24, no. 4, pp. 363–372, 1995.

[163] G. Younes, D. Asmar, and E. Shammas, "A survey on non-filter-based monocular Visual SLAM systems," *arXiv:1607.00470*, 2016.

[164] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendon-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 55–81, 2012.

[165] R. a. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 127–136, oct 2011.

[166] C. Fernandez-Carames, V. Moreno, B. Curto, J. F. Rodriguez-Aragon, and F. J. Serrano, "A real-time door detection system for domestic robotic navigation," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 76, no. 1, pp. 119–136, 2013.

[167] V. Dragoi, "Chapter 14: Visual processing: Eye and retina," in *Neuroscience Online, the Open-Access Neuroscience Electronic Textbook*, J. Concha, Ed. University of Texas Medical School at Houston, 1997, pp. 1–16.

[168] C. Foudil, D. Noureddine, C. Sanza, and Y. Duthen, "Path finding and collision avoidance in crowd simulation," *Journal of Computing and Information Technology*, pp. 217–228, 2009.

[169] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2, 2004, pp. 28–31.

[170] A. B. Chan, Z. S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008.

[171] S. Guo, Z. Qu, and L. Wang, "Camera pose estimation using frequency analysis," in *2014 International Conference on Information Science and Applications (ICISA)*, 2014, pp. 3–6.

[172] B. Zeisl, T. Sattler, and M. Pollefeys, "Camera pose voting for large-scale image-based localization," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2704–2712, 2015.

[173] Y. Do, "On the neural computation of the scale factor in perspective transformation camera model *," in *IEEE International Conference on Control and Automation (ICCA)*, 2013, pp. 712–714.

[174] W. L. Khoong, W. Y. Kow, H. T. Tan, H. P. Yoong, K. Teo, and K. Tze, "Kalman filtering based object tracking in surveillance video system," in *Proceedings of the 3rd CUTSE International Conference*, 2011.

[175] M. Hu, W. Hu, and T. Tan, "Tracking people through occlusions," *Proceedings - International Conference on Pattern Recognition*, vol. 2, pp. 724–727, 2004.

[176] G. Fechner, "Elements of Psychophysics," pp. Howes\DH\Boring\EG\–Ed, 1966.

[177] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions," *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pp. 1932–1939, 2009.

[178] A. Bhattachayya, "On a measure of divergence between two statistical population defined by their population distributions," *Bulletin Calcutta Mathematical Society*, vol. 35, pp. 99–109, 1943.